HeartGuard
Taking care of people heart's health with Machine Learning (Course Exam)

Sofia Tortolini
Digital Transformation Student
Course on Data Mining and Machine Learning
Campus of Cesena, 2022/2023
Email: sofia.tortolini@studio.unibo.it

Myriam Pollaccia
Digital Transformation Student
Course on Data Mining and Machine Learning
Campus of Cesena, 2022/2023
Email: myriam.pollaccia@studio.unibo.it

***Abstract*—The research of applying machine learning to the medical field is some of the most fascinating and complicated. Through this research it is possible, given a dataset, to apply a set of algorithms to be able to understand how accurately it is feasible to even estimate a chance of alerting a person with a certain probability for a heart problem. Estimates in the medical field are very sensitive, and finding actual, even just aggregated or anonymized data of people is difficult for privacy and for use in the private medical field. In this case for research purposes, a small dataset (of about 800 samples) was selected for the purpose of targeting an approach. The most complicated part turned out to be understanding, although never achieved with completeness, the parameters contained within it. Removing or replacing, or translating a column could be more or less "dangerous" to the medical field.**
The analysis went through 7 different machine learning and 1 deep learning algorithms, and for all of them, a little fine-tuning research to understand if the results could be improved. The conclusion was positive in directing the problem to machine learning solutions and allowed for finding materials and surveys as well as theses on the topic.

1. Introduction

"The year 2023 is the year of increased applications of artificial intelligence in heart researches. [Gio23]"

The application of AI and in particular machine learning to medicine is brimming with cases and benefits for society, people, and science. Recently an article [Gio23] was published highlighting that within a very few years, we have gone from looking up the symptoms of a tumor on a search engine to using "machine learning" algorithms to detect a heart attack on the fly, recognize "difficult" symptoms and choose the most appropriate treatment or procedure for a heart disease.
Furthermore, cardiovascular diseases are still the leading cause of death in our country, specifically more than 230,000 people die each year from ischemia, heart attack, heart disease, and cerebrovascular disease [Gas20].
In the 35-64 age group [GOV20], of fatal events, 30-40% die rapidly soon after the onset of symptoms and before arriving at the hospital. Longitudinal studies have shown that about half of coronary events are due to angina pectoris, which is rarely a reason for hospitalization.

Hence the interest in collecting and evaluating a similar sample of data on incidence, prevalence, and those symptoms or pathologies that could cause a heart attack.
The goal of this research arose from wondering whether it was possible given the normal parameters that come with heart examinations (in Italy), to apply some machine learning algorithms in order to estimate the probability of heart attack. Since medical data are particularly sensitive, the research has a substantial initial part to study and understanding them in the chosen datasets, looking for the best way to interpret them and operate accordingly.

1.1. Dataset research

Searching for a health dataset nowadays is not easy: some sources are mandatory-pay or do not allow export, unfortunately in this case for updated and recent data (2000-2023) nothing can be found. Kaggle through its in-depth library and community, on the other hand, opened us to several possibilities that however are not without problems:

- Heart Attack Analysis Prediction Dataset;
  This first dataset is of only 303 observations and already has all features converted from categorical to numerical, without an explanation of the strategies used. Once it was loaded through the Pandas library, however, it was noted that an ordinal encoding type was used, a choice that was not supported since in cases of some diseases the necessary condition of ordinal categorical data, meaning the categories have a natural order, was not true.

This dataset also has another .csv file that contains within it a large dataset of 3584 rows with only one column named ”98.6” that turned out to be a blood oxygen measurement, but since the rows did not match with the other file and it was not possible to know if and how that data had been extracted the search was shifted to another source.

- Healthcare Dataset on heart attack possibility;
  This dataset has the same values (303 rows) as the previous one but without the oxygen. It has similarly named value columns but with doubled end letters. A symptom that this is probably not a well-constructed dataset either, in addition to already having been converted numerically (still in ordinal mode) categorical features.

- Clevaland Dataset [Det+89];
  On the previous dataset there was a comment that led to a data file containing raw data, without columns names or other information, belonging to Uci Education Archives, which is indicating the authors, the provenance of the data and moreover the presence of missing values, confirmed by running

```
# Check for missing values in the DataFrame
missing_values = heart_data.isnull()
# Count missing values in each column
missing_counts = missing_values.sum()
```

  which returned a total of 6 missing values for 2 different column values. the data were already encoded in a numerical format and therefore was impossible to understand them, so at the end it was not chosen.

- Heart Failure Prediction - 2021 The research guided to this dataset, which is the one chosen to develop a solution and it’s made by 918 observations. In this dataset, 5 heart datasets are combined over 11 common features which makes it the largest heart disease dataset available so far for research purposes.

  With this last dataset, the investigation was initiated and went deeper into features from exploration to engineering.

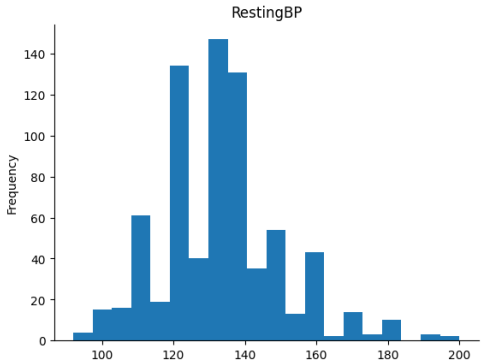
2. Data exploration

As explained, finding good datasets with properly defined attributes is a true challenge. In fact, even for the dataset chosen, only a brief description of every feature is available, without a real explanation of the meaning behind the values or the reason for them to be used in a heart disease inspection.
If, for some values like age and sex, everybody has kind of a general idea about the reasons why they might be of interest for heart disease detection, other attributes such as ST_slope are more complicated.

It has been decided to proceeded to go deep into the research, by looking at possible usage in other datasets or by trying to grasp some knowledge by medical research papers, also exploiting the research other students and researchers did in this field before us.

The overall features are:

- *Age: age of the patients [years]*
  Adults age 65 and older are said to be more likely than younger people to suffer from cardiovascular disease. Aging can cause changes in the heart and blood vessels that may increase a person’s risk of developing cardiovascular disease.
- *Sex: sex of the patient [M: Male, F: Female]*
  It is a common thing to say that women tend to develop heart disease 7-10 years later than men, and indeed the data reflects this thinking in a 75% sample of men compared to the total data present. But this is actually not always true [App11]: there is the possibility for women to be undertreated since general thought is for men to be more susceptible.
- *ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]*
  Angina is a chest pain caused by reduced blood flow to the heart muscles. It’s not usually life threatening, but it’s a warning sign that a person could be at risk of a heart attack or stroke.
  Typical (classic) angina chest pain consists of substernal chest pain or discomfort that is provoked by exertion or emotional stress and relieved by rest or nitroglycerine (or both).
  Atypical (probable) angina chest pain applies when two out of three criteria of classic angina are present.
  Non-specific chest pain is a classification where if less than one of the criteria of classic angina is present, symptoms are classified as non-specific [Her10].
- *RestingBP: resting blood pressure [mm/Hg]*
  Individuals with high blood pressure (higher than 120/80) are more susceptible to ischemic heart disease.



A problem faced with this data is the fact that blood pressure should contain two values, one for

systolic and one for diastolic pressure, but the dataset chosen shows only one. Considering the values are pretty high, the dataset is referring to systolic blood pressure, which, alone, cannot really be used as a reference.

- *Cholesterol: serum cholesterol [mm/dl]*
  High cholesterol makes it easier to develop fatty deposits in blood vessels. This makes it difficult for blood to flow through the arteries. If these deposits break apart, they can form a clot that leads to a heart attack or stroke.
  Even if it is stated that cholesterol was measured in mm/dl, cholesterol is actually always measured in mg/dl.
  Aside from that, the question might be if the serum cholesterol was intended to be HDL (high-density lipoprotein), also known as "good cholesterol" or LDL (low-density lipoprotein), also known as "bad cholesterol".
  The data seems to be referring to general cholesterol without discrimination between the two. This makes this data not really reliable for analysis, just like the considerations made for blood pressure.
  Moreover, even if the dataset is considered clean of outliers and null values, the reality is it contains 172 rows with a cholesterol value of 0, which is clinically impossible. Also, out of these, 150 have heartdisease=1, which results in a negative correlation between the serum cholesterol value and our target. At first, the presence of outliers was not apparent, so it was surprising to see this negative correlation, considering the common thought is that high cholesterol leads to heart disease, but it was initially justified with a research on the possibility of cholesterol not being a real determining factor [Pie20].
  Anyway, after noticing the outliers, the decision made was to drop the feature[Ese23], and not only did the negative correlation disappear (reinforcing the initial usual common thought of the two being positively correlated) but also the accuracy of the model slightly increased.
- *FastingBS: fasting blood sugar [1: if FastingBS >120 mg/dl, 0: otherwise]*
  High blood glucose can damage blood vessels and the nerves that control the heart. Over time, this can lead to disease.
  In this case, it is noted only if the values were higher than the threshold of 120 mg/dl of glucose in the blood at fast. An open point of discussion is why the same modality wasn't maintained for cholesterol.
- *RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of >0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]*
  In a resting electrocardiogram (ECG) it is possible to detect abnormalities including arrhythmia, and left ventricular hypertrophy.
  Specifically, for the dataset examined, there are three possible values: normal, ST and LVH.
  ST refers to the ST segment, which is the flat, isoelectric part of the ECG and it represents the interval between ventricular depolarization and repolarization (contraction and relaxation of the heart muscle).
  An abnormality of the ST segment (elevation or depression compared to normal values) is a signal of myocardial ischaemia or infarction.
  LVH, or left ventricular hypertrophy, can be detected through ECG as well.
  The point of insight was if the three values are ordinal in terms of criticity, in order to decide how to translate them in numerical values. Of course, ST and LVH values are worse than normal, but is LVH worse than ST? Since the full medical understanding in this case requires more knowledge on the topic, for the sake of the task it has been decided to treat the feature as not ordinal and go ahead with the research.
- *MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]*
  It has been shown that an increase in heart rate by 10 beats per minute is associated with an increase in the risk of cardiac death by at least 20%, and this increase in the risk is similar to the one observed with an increase in systolic blood pressure by 10 mm/Hg [Per09].
  A consideration must be taken into account: maximum heart rate is age-related information since the expected value is calculated by subtracting age from 220 [DP22], but considering some other studies, this value seems to not really be relevant to predict heart disease [San95].
- *ExerciseAngina: exercise-induced angina [Y: Yes, N: No]*
  Exercise increases the need for oxygen, which can result in pain.
  There were not proper studies concluding that patients with exercise-induced angina have a higher risk of heart disease, therefore investigating this feature might be interesting.
- *Oldpeak: oldpeak = ST [Numeric value measured in depression]*
  ST depression induced by exercise relative to rest.
  High values indicate higher stress on the heart [Wat00].
- *ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]*
  The maximal ST slope can be used reliably to predict the presence or absence and the severity of coronary artery disease in patients with anginal pain [Kar82].
  It is possible to clearly state that ST segments flat or downsloping higher than 1mm indicate an abnormality [Hil22], however, there are some contradictory

studies on the meaning of the upsloping [Pol06], so an analysis on the possible relationship with the target could be interesting.

- *HeartDisease: output class [1: heart disease, 0: Normal]*
  The presence of heart disease is the target of this research paper.

2.1. Data preparation

In agreement with what the previous section highlights, choices were made regarding the translation of categorical features. In particular, the request

```
complete_heartData[column].dtype == 'object'
```

returned 5 columns (Sex, ChestPainType, RestingECG, ExerciseAngina, and ST_Slope).

The research stuck with the following strategy:

- Sex and ExerciseAngina are simple binomial values (male and female - yes and no), therefore they have been translated with a "manual encoder"

```
binary_encoding = {"M": 0, "F": 1, "Y": 0, "N": 1}
complete_heartData["Sex"] =
complete_heartData["Sex"].map(binary_encoding)
complete_heartData["ExerciseAngina"] =
complete_heartData["ExerciseAngina"]
.map(binary_encoding)
```

- For the other columns the discussion for deciding to consider different types of encoders opens.

There are a few techniques for this purpose, depending on the nature of categorical data:

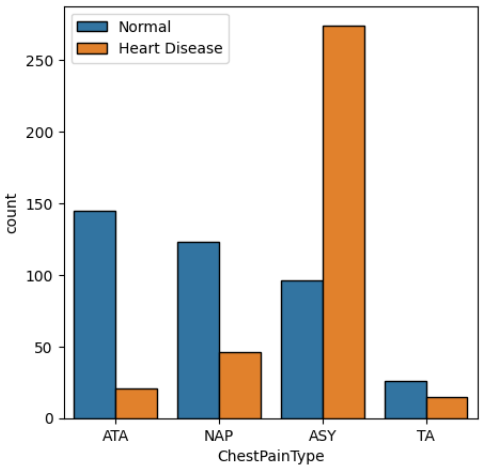
LE Label encoding is suitable when having ordinal categorical data, meaning the categories have a natural order. For example, "low," "medium," and "high" can be encoded as 0, 1, and 2, respectively;

OHE One-hot encoding is appropriate for nominal categorical data, where there is no inherent order among categories. It creates binary columns (0 or 1) for each category;

TE Target encoding is useful when having a categorical column, and the goal is to encode it based on the mean of the target variable for each category. This can be helpful when a correlation is suspected between the categorical variable and the target.

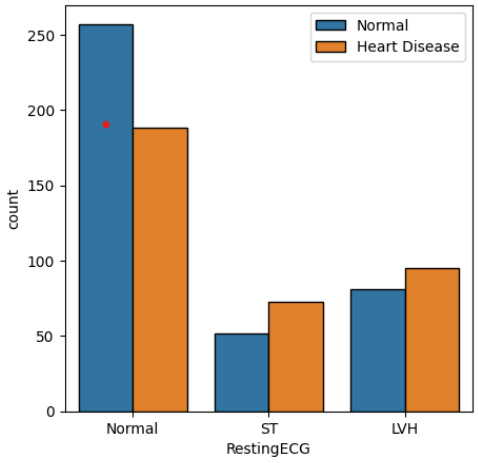
Before deciding, the importance of some of these columns on the target audience was investigated: based on the results, the types of translations and some other consequences changed.

In the case of ChestPainType, it does not seem to have any influence on the target outcome, and in support of this conclusion in the following graph

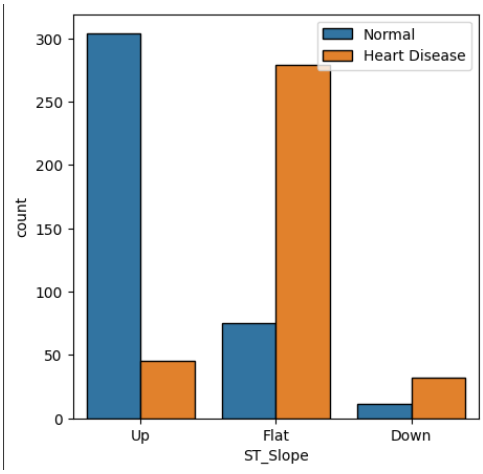


most cases are found to be correlated with ASY, which from the perspective of domain or knowledge about the data is associated with an asymptomatic patient. At the same time it is not possible to identify it with an ordinal translation since ATA or TA as values are not one of lesser importance than the other.

A similar argument applies in the case of the following RestingECG graph,



which again has no influence on the target, but is not ordinal either.



For the last value, on the other hand, there is a correlation with a flat value, which, however, determines an abnormality in the patient's coronary arteries.

At the end of this analysis, the ChestPainType, RestingECG, and ST_slope features were translated with the following target encoding:

```
encoder = ce.TargetEncoder(cols=["ChestPainType",
"RestingECG", "ST_Slope"])
encoder.fit(X_train, y_train)
X_train_encoded = encoder.transform(X_train)
X_test_encoded = encoder.transform(X_test)
encoded_features = pd.merge(X_train_encoded,
X_test_encoded, how="outer")
```

and in doing so, the dataset was cleaned and translated into numeric values, keeping in mind the best strategies to avoid errors.
Because the results of this analysis also resonated with research on the topic of which features could be retained or deleted, they were integrated into a series of points in the next chapter.

2.2. Feature analysis and selection

From the features' analysis of the dataset, it has been thought of selecting a subset of relevant features to consider for models.
For instance, as explained before, RestingBP (Resting Blood Pressure) only contains the systolic value, which is not enough. The same goes for the cholesterol value.
About MaxHR (Maximum Heart Rate), as explained before, the expected value is related to age (because of the calculation you have to make to get it). Also, considering the study found, it would be interesting to see if the feature is really relevant or not.
Investigating on ExerciseAngina is also interesting for the reasons explained in the paragraph and same goes for ST_Slope.

3. Model selection

The dataset was shuffled to be sure to not process data in order and then the division of train-set and test-set was made using the train_test_split from sklearn.model_selection. The division is not the default one in particular: test_size=0.15.

Deep learning models are best used on large volumes of data, while machine learning algorithms are generally used for smaller datasets like this case. In fact, using complex DL models on small, simple datasets culminates in inaccurate results and high variance - therefore that's why for this research the reader can find first the ML solutions and then one application of a DL algorithm.

3.1. Machine Learning

For evaluating the ML models in this paper classification metrics such as Accuracy, Precision, Recall, F1-score, and graph of confusion matrix as well as ROC-AUC have been chosen and calculated. in the code are implemented as functions so they can be called when needed:

```
def eval_model(y_test,y_pred):
  accuracy = accuracy_score(y_test, y_pred)
  precision = precision_score(y_test, y_pred)
  recall = recall_score(y_test, y_pred)
  f1 = f1_score(y_test, y_pred)

def eval_model_synthetic(y_test,y_pred):
  print(classification_report(y_test, y_pred))

def show_graph_ROCAUC(classifier,X_test, y_test):
  svc_disp = RocCurveDisplay
  .from_estimator(classifier, X_test, y_test)

def show_graph_confMATRIX(classifier,
                          X_test,y_test
                          ):
    disp = ConfusionMatrixDisplay.from_estimator()
```

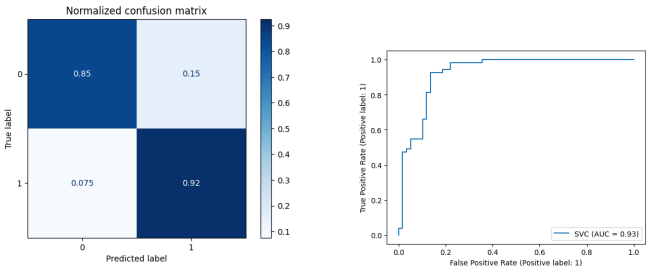
In this research paper we reported the graph and metrics that have been executed the last time the notebook developed was run. Metrics of course can sligthly change from one execution to the other, but the overall performance is stable.

In this paper everytime accuracy is mentioned, in reality, in the associated Colab Notebook there are also Precision, Recall and F1-score calculated.

3.1.1. SVM. Support vector machine (SVM) is a supervised learning method commonly used for classification. The initial trial with the SVM with default parameters was not good enough (accuracy: 0.63), therefore different Kernel functions were specified for the decision function until the optimal case was reached:

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear')
classifier.fit(X_train, y_train)
```

with Accuracy: 0.88.

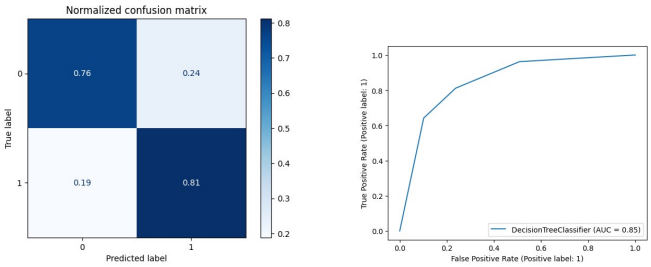


Confusion Matrix ROC curve

Figure 1: SVM Classifier

3.1.2. Decision Tree Classifier. A decision tree is a flowchart-like tree structure where an internal node represents a feature, the branch represents a decision rule, and each leaf node represents the outcome.
The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in a recursive manner called recursive partitioning.
Of course also in this case the accuracy of the standard algorithm on Heart Data was not so much performing: accuracy was at 0.78.

For fine-tuning a Decision Tree, first of all, the criterion that allows to use the different attribute selection measure was changed from "gini" to "entropy" and this increased the performance to 0.82, which was good.
But as the algorithm before, RandomizedSearchCV has been tried to find a number for the max_depth parameter without using the default one. This tuning didn't help to increase the



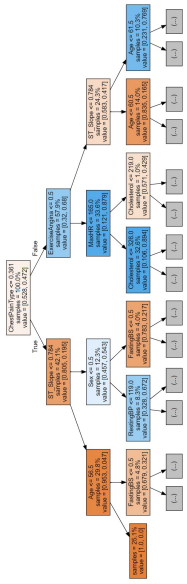
Confusion Matrix ROC curve

Figure 2: DT Classifier

accuracy of the model which, after different trial, remains pretty much the same, therefore it has been kept the final value of 0.82 as a mean.
Since this was not performing as expected, the approach chosen was to try with multiple decision trees created using different random subsets of the data and features, which is exactly what a Random Forest Algorithm does.

3.1.3. Random Forest. In this algorithm each decision tree is like an expert, providing its opinion on how to classify the data. Predictions are made by calculating the prediction for each decision tree, then taking the most popular result.

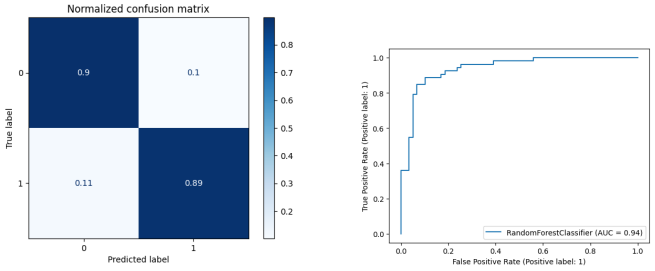
From the very first execution, the algorithm was performing really well with an accuracy of 0.87, definitely better than the previous one.
Using the library graphviz was also possible to show the trees and help the visualization and learning of the internal process of the algorithm:



Moreover, thanks also to RandomizedSearchCV library, it was possible to try a fine-tuning of parameters, in particular in this case, the following:

```
param_dist = {'n_estimators': randint(50,500),
              'max_depth': randint(1,20)}
```

The parameter max_depth determines the maximum depth of individual trees in the ensemble, influencing the model's ability to capture intricate patterns in the data. Setting an appropriate max_depth prevents overfitting by controlling the complexity of the trees.
On the other hand, n_estimators define the number of trees in the forest, and finding the right balance is vital. Too few trees may result in underfitting, while too many may lead to increased computational costs without significant improvement in predictive accuracy.



Confusion Matrix ROC curve

Figure 3: RF Classifier

By fine-tuning these parameters, the research strikes a balance between model complexity and generalization with

an increase in the accuracy to 0.89.

3.2. kNN

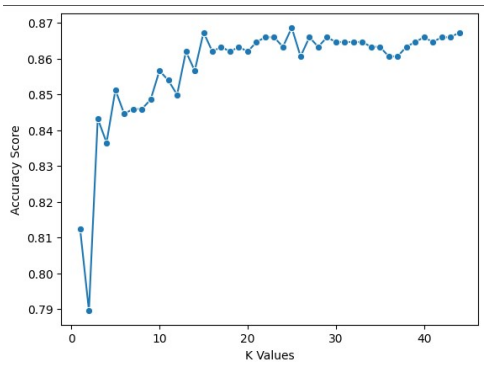
The kNN algorithm can be considered a voting system, where the majority class label determines the class label of a new data point among its nearest 'k' (where k is an integer) neighbors in the feature space.
One thing important for implementing this algorithm is to normalize features:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

After this passage, the execution with standard parameters gave a result of 0.83, which is pretty high.
This time, instead of using RandomizedSearchCV the research focused on the number of n_neighbors: selecting an appropriate value for n_neighbors is crucial as it directly influences the model's sensitivity to local patterns in the data. A smaller n_neighbors value results in a more flexible, but potentially noisy, model, while a larger value can lead to oversmoothing and overlooking subtle patterns. So, it has been decided to proceed with this strategy:

```
k_values = [i for i in range (1,45)]
scores = []
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    score = cross_val_score(knn, X, y, cv=5)
    #trial with different values and
    #then added to scores array
    scores.append(np.mean(score))
#and then print the results
sns.lineplot(x = k_values,
             y = scores,
             marker = 'o')
plt.xlabel("K Values")
plt.ylabel("Accuracy Score")
```

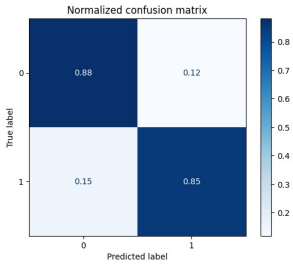
Which results in a graph to help the right picking of the value.



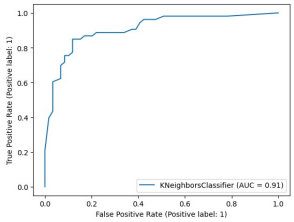
The optimal k value we think is therefore around 25, and this resulted in an accuracy of 0.87.

3.3. Boosting Classifier

Unlike standalone models that have been implemented since this paragraph, boosting techniques sequentially build



Confusion Matrix



ROC curve

Figure 4: kNN Classifier

an ensemble of weak learners, each focusing on the short results of its predecessors.
AdaBoost, for instance, assigns different weights to misclassified instances, emphasizing their importance in subsequent iterations. On the other hand, XGBoost, an extension of gradient boosting, employs a gradient-based optimization approach.
AdaBoost - In the very first approach, default parameters were chosen, as usual in this research paper, and the result was higher than expected with an accuracy of 0.86. AdaBoost uses Decision Tree classifier as the default classifier, which in the previous case was giving a medium accuracy result of 0.82.
However, since the scope of the research is to investigate more classifiers and different solutions, it has been decided to combine other classifiers: SVM and Random Forest to see the Ada results.

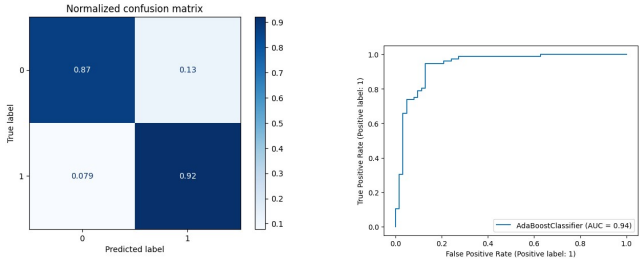
```
svm = SVC(kernel = "linear") #sub_estimator1
rf = RandomForestClassifier(max_depth= 4,
                n_estimators= 409) #sub_estimator2

abc1 =AdaBoostClassifier(n_estimators=50,
estimator=svm, learning_rate=1,algorithm='SAMME')
abc2 =AdaBoostClassifier(n_estimators=50,
estimator=rf, learning_rate=1)
```

It's important to notice the change in the algorithm to "SAMME", this happened because estimators must support the calculation of class probabilities and this is not the case for SVM. The results obtained were the following, the first print is for SVM, the second for RF:

	precision	recall	f1-score	support
0	0.76	0.76	0.76	62
1	0.80	0.80	0.80	76
accuracy			0.78	138
macro avg	0.78	0.78	0.78	138
weighted avg	0.78	0.78	0.78	138

	precision	recall	f1-score	support
0	0.90	0.87	0.89	62
1	0.90	0.92	0.91	76
accuracy			0.90	138
macro avg	0.90	0.90	0.90	138
weighted avg	0.90	0.90	0.90	138

As for the other models ROC-AUC curves and Confusion Matrix were printed, but just for the second model:



Confusion Matrix ROC curve

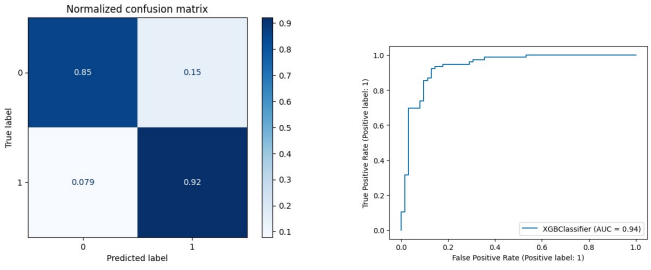
Figure 5: AdaBoost Classifier

can demonstrate an increase in the accuracy up to 0.90.

XGBoost - was the second boosting algorithm implemented. The first trial with the default parameter was higher than the optimized version of SVM and DTC, with an accuracy of 0.88 (mean).
With GridSearchCV it was possible to change (try) different parameters in particular: the 'learning_rate' which determines the step size (at each iteration) while moving towards a minimum of the loss function; then 'max_depth' which controls the maximum depth of individual trees - influencing their complexity and the model's ability to capture intricate patterns; finally, 'subsample' parameter, which defines the fraction of training instances used for growing trees, impacting the model's generalization capability [Art23].

```
    param_grid = {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.1, 0.01, 0.001],
    'subsample': [0.5, 0.7, 1]
}
```

resulting in a model (with the best parameters) with an accuracy of 0.89.



Confusion Matrix ROC curve

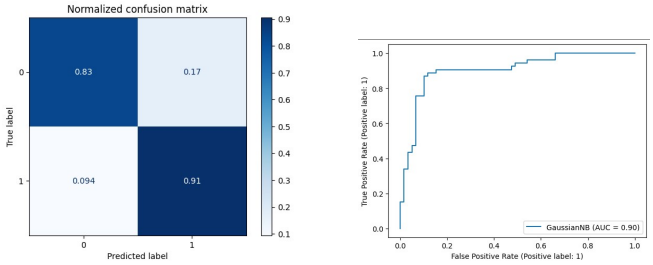
Figure 6: XGBoost Classifier

3.4. Gaussian Naive Bayes

The last machine learning classification algorithm was the Gaussian NB, rooted in Bayes' theorem. This algorithm assumes that features are conditionally independent given the class labels, simplifying the modeling process. Despite its seemingly naive assumption, Gaussian Naive Bayes proves to be remarkably effective, particularly in scenarios where the distribution of features within each class follows a Gaussian (normal) distribution.
From the default configuration which gave a 0.86 on accuracy as the KNN algorithm, GridSearchCV was used to test parameters-tuning as before.

```
    param_grid = {
    'var_smoothing': [1e-9, 1e-8, 1e-7
    , 1e-6, 1e-5],
    'priors': [None, [0.5, 0.5]]
}
```

In this particular case: 'var_smoothing' parameter is a smoothing hyperparameter added to the variances of each feature to prevent issues arising from zero division, ensuring numerical stability during probability calculations; 'prior' parameter, instead, allows users to specify prior probabilities for the classes. While Gaussian Naive Bayes typically assumes uniform class priors, adjusting the 'prior' parameter becomes relevant when dealing with imbalanced datasets.
But actually, this time, the result was equal to the value before, without an improvement in the overall performances, this means the default parameters, probably were already the optimal ones.



Confusion Matrix ROC curve

Figure 7: Gaussian Naive Bayes Classifier

3.5. Deep Learning

This section opens a well-known discussion: when is it best to apply machine learning and when is it best to apply deep learning?

Element	Machine Learning	Deep Learning
Data	**Large** data (~ hundreds)	**Huge** data (~ thousands)
Accuracy	**High** accuracy	**Best** accuracy (esp. high-dimensional data)
Training time	**Less** time (~minutes)	**Large** Time (~hours, days)
Required hardware	CPU	GPU
Features	Manual	Learned
Interpretability	Good	Low

During classes, this was an open point and the fact is that the answer can be predetermined before.
For the sake of this research paper, the goal was to demonstrate this, plus the fact that sometimes is not true

that applying one approach is better than the other.

Multilayer Perceptron (MLP) classifier - It consists of multiple layers of interconnected artificial neurons, also known as perceptrons. These layers typically include an input layer, one or more hidden layers, and an output layer. Each neuron in the network receives input signals, applies a non-linear activation function, and passes the transformed output to the next layer. This process continues until the final layer, which produces the classification output.
The result of a standard execution of MLP produces an accuracy (after 130 iterations) of 0.87, which was perfectly in-line with SVM, KNN and XGB.
However, this time with GridSearchCV we tried different params for the solver and the activation layer:

```
grid = {'solver': ['lbfgs', 'sgd', 'adam'],
'activation': ['identity', 'logistic',
'tanh', 'relu']}
#with gridsearch we can try different paramether
#for the solver and for the activation,
#instead for the others we can set
#them differently from before
clf_cv = GridSearchCV(
    MLPClassifier(random_state=1, max_iter=5000,
    hidden_layer_sizes=(3,3), alpha=1e-5),
    grid, #the parameter to search
    n_jobs=-1, #is to define how many CPU cores
    #of computer to use (-1 is for all
    #the cores available)
    cv=10 #is the number of splits for
    #cross-validation
    )
```

With this configuration the research ended up with an accuracy (optimized) of 0.89, actually lower than RF or Ada algorithms.

	precision	recall	f1-score	support
0	0.89	0.92	0.90	59
1	0.90	0.87	0.88	53
accuracy			0.89	112
macro avg	0.89	0.89	0.89	112
weighted avg	0.89	0.89	0.89	112

However, to conclude this part it's important to report this dataset: Cardiovascular Disease dataset, which with 70000 rows it seems perfect for a further Deep Learning analysis, but it didn't seem appropriate to work on it for the lack of information about the source of the data and for some issues with blood pressure values and strange correlations between features.

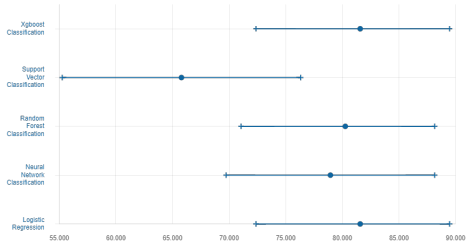
4. Conclusion

The overall performances of the algorithms and the demonstration of a machine learning approach developed on this problem are the following (summarized):

TABLE 1: Algorithms' Result

Name	Accuracy Score (mean)
Support Vectore Machine	0.88
Decision Tree Classifier	0.82
Random Forest	0.89
KNN	0.87
Ada Boost	0.90
XGBoost	0.89
DL: Multilayer Perceptron	0.89

It's interesting to compare these results, with the ones stated on the websites of the dataset itself, where there is this chart:



It's possible to say that this paper is perfectly in line with the best results of some of the stated algorithms on the chart. In fact, with regards to the Boost algorithm even for this paper, it's true the best performance obtained (in this case AdaBoost was the best, that is not even displayed there); the SVM of this paper performs better than the versions stated in the chart (with a mean of 0.88), so probably it means that with this research it has been found the best parameter tuning. Then, also for the DL approach, we can see that our result is on the top best score (0.89).

For this particular dataset the best ML algorithms are the boosted ones, to support this conclusion there is also this guide, in which the author utilized a CatBoostClassifier.

Since the beginning we mentioned some strategies to enhance even more the capability of these algorithms to increase in accuracy, here it's time to say that there is actually an entire thesis to support the thoughts in this paper: *Investigating Heart Disease Datasets and Building Predictive Models* [II21] that has been briefly analyzed to discover if the initial points were the correct ones.

The importance of research in this field seems undeniable, just like the infinite possibilities of machine learning applications. The main issue here lies in data.
Dozens of notebooks and other research papers has been found, but the common thread is the terrible care of the datasets we have to deal with.
For this matter, there is actually another entire thesis about the lack of a proper dataset to work with [Sim21].
Some of the datasets that have not been considered (for the reasons already explained) actually contained really interesting features not present in the one chosen, such as alcohol consumption, smoking, thalassemia, number of major vessels resulting from a thallium test and so on.
Investigating on the many values patients usually get checked on could lead to interesting discoveries, maybe

leading to dismantling already consolidated theories.
What could happen if these techniques and the same assumption would be applied, for instance, to Alzheimer's disease research?

Having a proper dataset with all of these data and with a real clear understanding of the meaning behind it (which, again, was one of the main issues that has been faced) could really make a difference in heart disease prediction and, in general, on public health for the entire world.

References

[Kar82] M. Kardash. *The slope of ST segment/heart rate relationship during exercise in the prediction of severity of coronary artery disease*. https://academic.oup.com/eurheartj/article-abstract/3/5/449/424605. Oct. 1982.

[Det+89] Robert C. Detrano et al. "International application of a new probability algorithm for the diagnosis of coronary artery disease." In: *The American journal of cardiology* 64 5 (1989), pp. 304–10. URL: https://api.semanticscholar.org/CorpusID:23545303.

[San95] Leiv Sandvik. *Heart rate increase and maximal heart rate during exercise as predictors of cardiovascular mortality: a 16-year follow-up study of 1960 healthy men*. https://pubmed.ncbi.nlm.nih.gov/8574463/. Aug. 1995.

[Wat00] Takuya Watanabe. *Exercise-induced ST-segment depression: imbalance between myocardial oxygen demand and myocardial blood flow*. https://pubmed.ncbi.nlm.nih.gov/10707755/. Feb. 2000.

[Pol06] George Polizos. *The Value of Upsloping ST Depression in Diagnosing Myocardial Ischemia*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6932726/. July 2006.

[Per09] Christine Perret-Guillaume. *Heart rate as a risk factor for cardiovascular disease*. https://pubmed.ncbi.nlm.nih.gov/19615487/. Aug. 2009.

[Her10] Luke K. Hermann. *Comparison of frequency of inducible myocardial ischemia in patients presenting to emergency department with typical versus atypical or nonanginal chest pain*. https://pubmed.ncbi.nlm.nih.gov/20494662/. Apr. 2010.

[App11] A.H.E.M. Maas Y.E.A. Appelman. *Gender differences in coronary heart disease*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3018605/. May 2011.

[Gas20] Fondazione Cardiovascolare De Gasperis. *Malattie cardiovascolari prima causa di morte in Italia*. https://www.degasperis.it/malattie-cardiovascolari-prima-causa-di-morte-in-italia.html. Feb. 2020.

[GOV20] Salute GOV. "La situazione sanitaria del paese". In: *https://www.salute.gov.it/imgs/C_17_navigazioneSecondariaRelazione_1_listaCapitoli_capitoliItemName_1_scarica.pdf* (2020).

[Pie20] Ann Pietrangelo. *Does High Cholesterol Cause Heart Disease?* https://www.healthline.com/health/cholesterol-and-heart-disease#prevention/. May 2020.

[II21] Brandon Simmons II. "Investigating Heart Disease Datasets and Building Predictive Models". In: *Graduate Faculty of Elizabeth City State University* I (2021). URL: https://libres.uncg.edu/ir/ecsu/f/Brandon_Simmons_Thesis-Final.pdf.

[Sim21] Brandon Simmons. *Investigating Heart Disease Datasets and Building Predictive Models*. https://libres.uncg.edu/ir/ecsu/f/Brandon_Simmons_Thesis-Final.pdf. May 2021.

[DP22] Centers for Disease Control and Prevention. *Target Heart Rate and Estimated Maximum Heart Rate*. https://www.cdc.gov/physicalactivity/basics/measuring/heartrate.htm. June 2022.

[Hil22] Jonathan Hill. *Exercise tolerance testing*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1123032/. May 2022.

[Art23] Medium Article. *Optimizing XGBoost: A Guide to Hyperparameter Tuning*. https://medium.com/@rithpansanga/optimizing-xgboost-a-guide-to-hyperparameter-tuning-77b6e48e289d. Jan. 2023.

[Ese23] Berkay Eser. *Applying several methods to cholesterol feature*. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction/discussion/438385. Sept. 2023.

[Gio23] Il Giorno. *Quando l'algoritmo salva dall'infarto: il cuore si cura con l'intelligenza artificiale*. https://www.ilgiorno.it/salute/cardiologia-intelligenza-artificiale-cwaj26gy. Oct. 2023.