Machine Learning Engineer Nanodegree

Capstone Project

Tatiana Zolhof Panisset
December 28th, 2018

I. Definition

Project Overview

Heart Disease is a global problem that takes lives of millions of people every year. According to the World Health Organization - WHO[1], 17.9 million people die every year from cardiovascular diseases, corresponding to 31% of all deaths worldwide. In the United States, data taken from the Center for Disease Control and Prevention – CDC website[2] show that about 610,000 people die of heart disease in the country every year, corresponding to 1 in every 4 deaths. Also according to the WHO, triggering these diseases are tobacco smoking, unhealthy diet, physical inactivity, and the harmful use of alcohol. These in turn show up in people as raised blood pressure, elevated blood glucose, and overweight/obesity.

This is a subject of constant research, in order to identify actions that might reduce the overwhelming numbers aforementioned. Academic research references can be found in the UCI – Machine Learning Repository[3], which contains several papers on the theme.

The goal of this project is to study a dataset of people, with information about their characteristics and health condition, and use Machine Learning techniques to create a model to predict the presence of heart disease in them, and also in new, unseen data. This would allow to take action to prevent the evolution of the condition, when it is already present, but also maximize the awareness of the problem as a whole, encouraging prevention efforts.

Problem Statement

Using the Heart Disease Data Set from the UCI – Machine Learning Repository[4], which contains several attributes for a group of people, it is necessary to create a model to predict the presence of heart disease in them that can also be generalized to data not contained in the original dataset. This is a Supervised Learning Classification problem, since there is a defined set of features and corresponding labeled outputs, for which there are many algorithms and metrics that can be applied.

The strategy to solve this problem starts with extensively exploring the dataset, to gather insights from the data and make any preprocessing steps that are needed. The next step will be building the Machine Learning models, trying several options – including Decision Trees, Ensemble Methods, Support Vector Machines, and Logistic Regression - to see the one that brings up the best performance. Having chosen the model, it will be necessary to tune it, using grid search for different parameters. The last step will be gathering the results, according to the metric detailed in the section below, and comparing them to the benchmarks obtained by previous studies.

Metrics

Since this is a classification problem, and also considering the benchmarks that the project's results will be compared to, accuracy will be the evaluation metric of the model. This is defined by:

$$\text{accuracy} = (\#\text{ of correctly classified examples} / \text{total } \#\text{ of examples}) * 100$$

II. Analysis

Data Exploration

About the dataset

The Heart Disease dataset contains 14 attributes for a group of people, including numerical and categorical variables, as follows:

- ‘age’: age (numerical);
- ‘sex’: sex (categorical);
- ‘cp’: chest pain type (categorical);
- ‘trestbps’: resting blood pressure (numerical);
- ‘chol’: serum cholesterol in mg/dl (numerical);
- ‘fbs’: fasting blood sugar (categorical);
- ‘restecg’: resting electrocardiographic results (categorical);
- ‘thalach’: maximum heart rate achieved (numerical);
- ‘exang’: exercise induced angina (categorical);
- ‘oldpeak’: ST depression induced by exercise relative to rest (numerical);
- ‘slope’: the slope of the peak exercise ST segment (categorical);
- ‘ca’: number of major vessels colored by fluoroscopy (categorical);
- ‘thal’: thallium stress test (categorical);
- ‘pred’: predicted attribute (categorical).

It is important to mention that the original dataset contains 76 attributes; however, 14 of them were found most helpful and adopted in past studies - including those taken as benchmarks here. For this reason, and also considering that the 14 variables cover the characteristics and health conditions related to heart disease, this is the dataset used in the project.

A sample of the first 5 lines of the data set is provided below:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	pred
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.0	6.0	0
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	2
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.0	7.0	1
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	0
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.0	3.0	0

Dataset statistics

The dataset has 303 observations, with the following statistics:

Stat \ Var	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	pred
mean	54.44	0.68	3.16	131.69	246.69	0.15	0.99	149.61	0.33	1.04	1.6	0.66	4.72	0.94
std	9.04	0.47	0.96	17.6	51.78	0.36	0.99	22.87	0.47	1.16	0.62	0.93	1.94	1.23
min	29.0	0.0	1.0	94.0	126.0	0.0	0.0	71.0	0.0	0.0	1.0	0.0	3.0	0.0
25%	48.0	0.0	3.0	120.0	211.0	0.0	0.0	133.5	0.0	0.0	1.0	0.0	3.0	0.0
50%	56.0	1.0	3.0	130.0	241.	0.0	1.0	153.0	0.0	0.8	2.0	0.0	3.0	0.0
75%	61.0	1.0	4.0	140.0	275.0	0.0	2.0	166.0	1.0	1.6	2.0	1.0	7.0	2.0
max	77.0	1.0	4.0	200.0	564.0	1.0	2.0	202.0	1.0	6.2	3.0	3.0	7.0	4.0

The statistics above show the mean, standard deviation, minimum, maximum and the percentiles of the data. Those summarize central tendency, dispersion and shape, being specially useful for numerical variables – revealing, in particular, quite different ranges that may require scaling in the preprocessing step.

However, not all of those measurements make sense for the categorical variables of the data. Hence, it is also important to include the median as a central tendency statistic:

Stat \ Var	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	pred
median	56.0	1.0	3.0	130.0	241.0	0.0	1.0	153.0	0.0	0.8	2.0	0.0	3.0	0.0

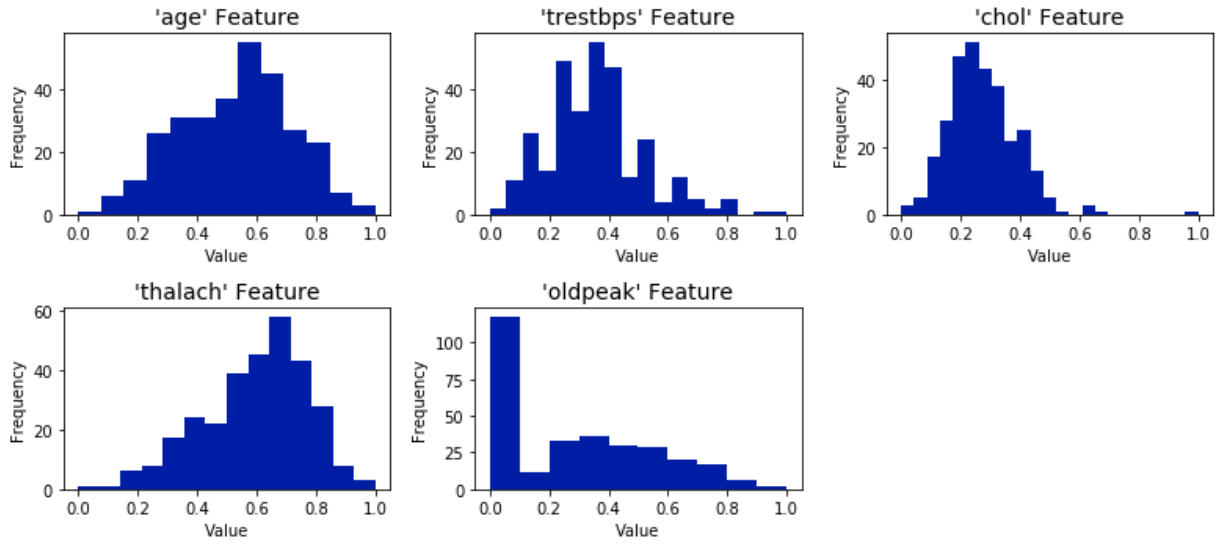
Abnormalities / interesting qualities about the data

The abnormalities / interesting qualities about the data that needed to be discussed and addressed (in the section Data Preprocessing, below) are:

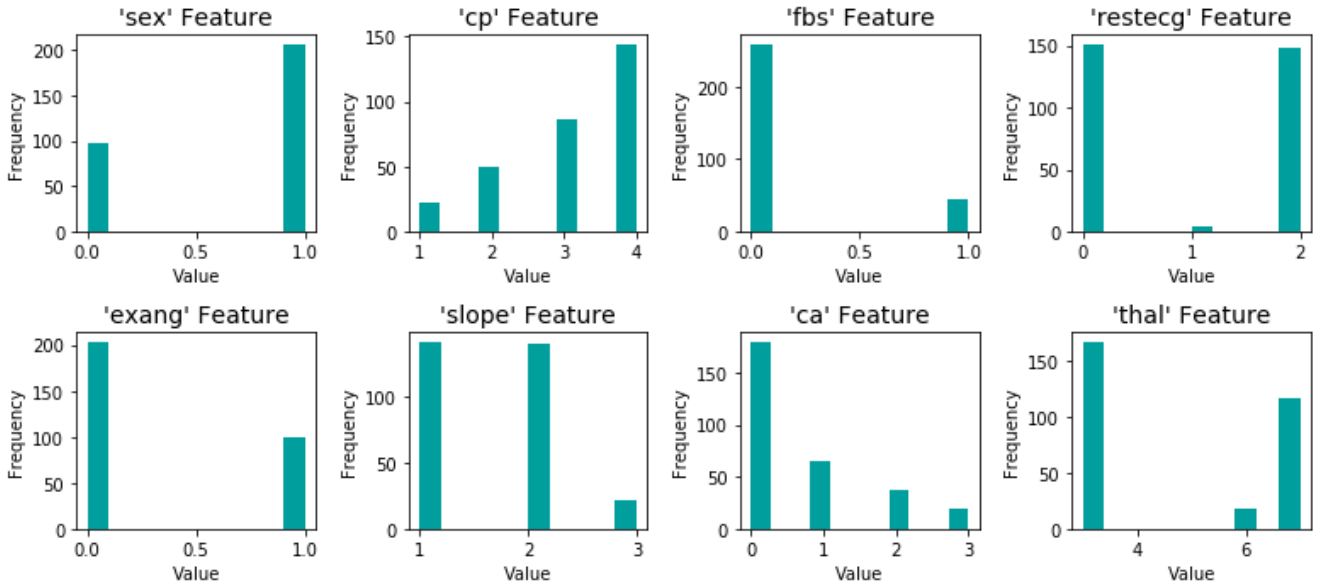
- There are 6 observations in the data base containing missing values, all of them present in the categorical variables ‘ca’ and ‘thal’;
- The outcome variable is integer valued from 0 (no presence of heart disease) to 4. The experiments previously made on the dataset, which will be used as benchmarks, have concentrated on attempting to distinguish presence (values 1,2,3,4) from absence (value 0) of heart disease. Thus, this mapping needs to be done;
- The ‘oldpeak’ feature has a right-skewed distribution;
- There are a few outliers for the variables 'trestbps' (9 observations), 'chol' (5 observations), and 'thalach' (1 observation). Those were identified by the Tukey's Method: an outlier step was calculated as 1.5 times the interquartile range (IQR), and a data point with a feature that was beyond an outlier step outside of the IQR for that feature was considered abnormal.

Exploratory Visualization

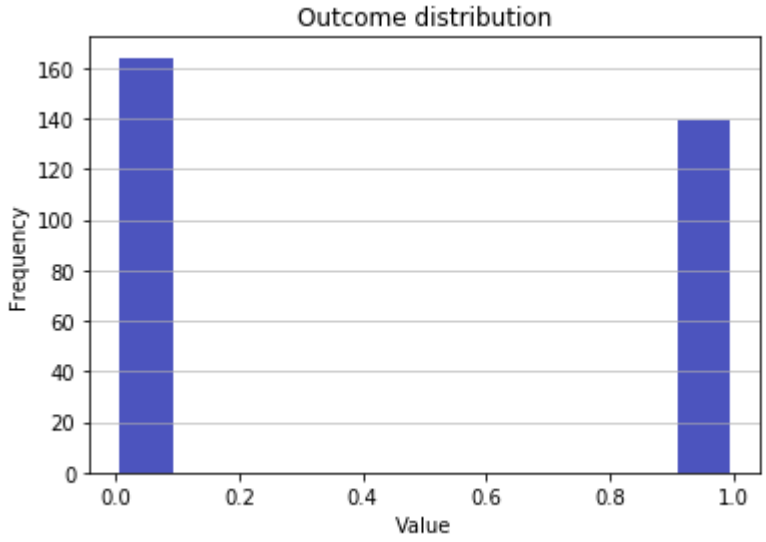
The first visualization shows the distribution of the numerical variables, in order to identify the need of data transformation. As can be seen below, the distributions for ‘age’, ‘trestbps’, ‘chol’ and ‘thalach’ are approximately normal, but the distribution for the ‘oldpeak’ feature is right-skewed:



The second visualization shows the distribution of the categorical features. In this case, the goal was to confirm that there were no columns with the same value – if so, those could be excluded from the dataset since they would not be adding any useful information:



The last visualization intended to verify if the output classes were imbalanced – considered, here, as the case in which there's a minority class of 10% or 20%[5]. The plot shows that the classes have a similar number of occurrences on the dataset; hence, no additional action is needed. Also, since the classes are not imbalanced, the use of accuracy as a metric to assess the Machine Learning models that will be tested is appropriate.



Algorithms and Techniques

Five algorithms will be tested with the dataset; all of them chosen for being classic approaches to the classification problem investigated in the project. They also have specific advantages, respectively:

- Logistic Regression: simple to interpret; can be regularized to avoid overfitting;
- Decision Tree Classifier: simple to interpret and visualize; can handle both numerical and categorical data;
- Adaboost Classifier: being an ensemble method, one of its advantages is the good performance and accuracy of the results; not prone to overfitting;
- Support Vector Machine: allows fitting non-linear decision boundaries;
- Random Forest Classifier: easy to use algorithm; the default hyper-parameters often produce a good prediction result; not prone to overfitting.

The techniques that will be used are the following:

- Splitting the dataset in training and testing sets so it is possible to make the evaluation of the final model in a different set than the one it was trained on;

- K-fold cross-validation, as a technique to make better use of the data available for the learning task. In order to compare the aforementioned algorithms, it is necessary to create a validation set to the data in which the results will be calculated and compared. However, this can impact the amount of data available to train the model, especially when there is not a very large number of observations. While providing better use of our data, K-fold validation also addresses the problem of having a particular combination of training/validation sets impacting the conclusions since it uses the average of the results obtained on the different groups;
- Grid Search for tuning the parameters of the chosen algorithm;
- Fixed random_state variable so all the steps of the project can be reproduced at any time.

Benchmark

The 'Past Usage' section of the 'heart-disease.names' file contained on the UCI - Machine Learning Repository[6] shows results of studies made with this dataset, which obtained an accuracy between 74.8 and 78.9%. Hence, those will be the benchmarks with which the project's results will be compared to.

III. Methodology

Data Preprocessing

At this step, the previously documented abnormalities / interesting qualities about the data, as well as some other adjustments required for the dataset, were discussed and addressed as follows:

Abnormalities / interesting qualities

- Missing values: since there were only 6 observations with missing values, and those showed in categorical variables ('ca' and 'thal'), they were replaced by the median of the respective column;

- Outcome variable: to be coherent with the studies taken as benchmarks for this project, the values 1, 2, 3 and 4 of the outcome variable were mapped to 1, indicating the presence of heart disease. The 0 value indicates its absence;

- ‘oldpeak’ feature: because it had a right-skewed distribution, a log-transformation was applied so that the very large and very small values would not negatively affect the performance of the learning algorithms;

- Outliers: considering that 1) only a few observations were categorized as outliers; 2) those observations were not common among the features; 3) the dataset is not very large (so that every observation is important, and should not be easily discarded); and 4) no clustering distance-based algorithms was going to be used (those are specially affected by outliers), these observations were kept for the next steps; however, it was necessary to observe/monitor any impact they might bring to the algorithms' performance.

Additional adjustments

The dataset statistics showed that numerical variables had quite different ranges. Because of that, in addition to the previous transformations, the numerical features were scaled, ensuring that each feature was in the 0-1 range and was treated equally when applying the supervised learners.

Finally, the categorical features were one-hot encoded so they could be appropriately treated by the algorithms. The final set of features, including numerical and categorical variables, had 28 columns.

Implementation

The implementation of the project used, for the most part, the tools provided by Python’s libraries like sklearn, pandas, numpy, and matplotlib. In some cases, simple functions were coded to support additional tasks.

For the preprocessing steps, the following functions were used:

- data.fillna(data.median()), to replace the missing values for the respective column’s median;

- data.describe, to get the dataset statistics;

- matplotlib.pyplot’s hist, to plot the variables’ distributions;
- lambda function, to do the logarithmic transformation of the feature ‘oldpeak’;
- sklearn.preprocessing’s MinMaxScaler, to perform feature scaling;
- pandas’ get_dummies, to one-hot encode categorical variables;
- numpy’s percentile, to implement the Tukey's Method for finding outliers;
- sklearn.model_selection’s train_test_split, to create the training and testing sets with the proportion of 80% and 20%, respectively.

The random_state variable was set to 42 along the project to ensure the results’ reproducibility, as mentioned before.

For building the Machine Learning models, the strategy was to use sklearn.model_selection’s Kfold and cross_val_score to create a set of 5 folds / splits for the training data, in which the different algorithms were trained. Then, using the accuracy as the metric, the results for each run of the cross validation were averaged.

The models were imported from sklearn as well – sklearn.linear_model (Logistic Regression), sklearn.tree (Decision Tree Classifier), sklearn.ensemble (Adaboost Classifier and Random Forest Classifier), and sklearn.svm (Support Vector Machine). At this step, the models were trained using their default parameters so that the tuning could be made for the best choice on the refinement step.

Refinement

Considering the accuracy metric, Logistic Regression was the model with best result. Hence, the next step was to refine / improve its performance, considering that the accuracy initially obtained for it was 83.49%.

The refinement was focused on regularization parameters. Training a model such as a Logistic Regression means choosing parameters that give the best fit to the data, by minimizing the error between what the model predicts for the outcome variable compared to what the outcome variable actually is.

However, when doing this, the model might tailor the parameter values to peculiarities in the data, fitting it almost perfectly, but performing poorly in future

data where these peculiarities don’t appear. Regularization, then, comes into place by applying a penalty to increasing the magnitude of the model’s coefficients, adjusting them to prevent overfitting.

With the help of the Grid Search tool, imported from sklearn.model_selection, the following parameters were tested:

- 'penalty' – lasso and ridge regularization: 'l1', 'l2'. Lasso Regression (l1) adds the absolute value of magnitude of coefficients as penalty term, while Ridge regression (l2) adds the squared magnitude of the coefficients as penalty term;
- 'C': 0.001, 0.01, 0.1, 1, 10, 100. This also controls model complexity; as the magnitude of the model’s coefficients increase, there will be an increasing penalty depending on the regularization type l1 or l2 as well as the magnitude of the C parameter.

Grid Search showed that the best combination of the hyper-parameters above was ‘penalty’ = ‘l2’, and ‘C’ = 0.1. Using those numbers, the final accuracy for the training set improved to 85.12%. It was also possible to calculate the accuracy for the testing set: the value obtained was 90.16%, a promising result that will be better analyzed and compared with the chosen benchmarks in the following sections.

IV. Results

Model Evaluation and Validation

The final model for the project was Logistic Regression, with hyper-parameters ‘penalty’ = l2, and ‘C’ = 0.1, leading to a final accuracy of 90.16% for the testing set. This architecture was chosen because it showed the best results when compared to four other models and several options of parameters:

Model	Accuracy(%)
Logistic Regression - untuned (training set)	83.49
Decision Tree - untuned (training set)	69.47

Adaboost - untuned (training set)	75.22
Support Vector Machine - untuned (training set)	83.48
Random Forest - untuned (training set)	77.70
Logistic Regression – tuned (training set)	85.12
Logistic Regression – tuned (testing set)	90.16

It is important to mention that Logistic Regression is one of the most used Machine Learning algorithms for classification. It is a simple and robust algorithm that perform well enough in many tasks, specially when combined with actions like regularization and careful dataset preprocessing.

Also, the fact that K-Fold cross-validation was used in every step of the way helped assure not only that the dataset was used on its full capacity, but also that the training/validation process took place with several combinations of the input data, avoiding that specific situations and groups could impact the results.

Finally, since this is a health related project, one additional evaluation of the results is to get the following metrics:

- recall, or sensitivity, defined as the proportion:

  recall = (# of true positive examples/

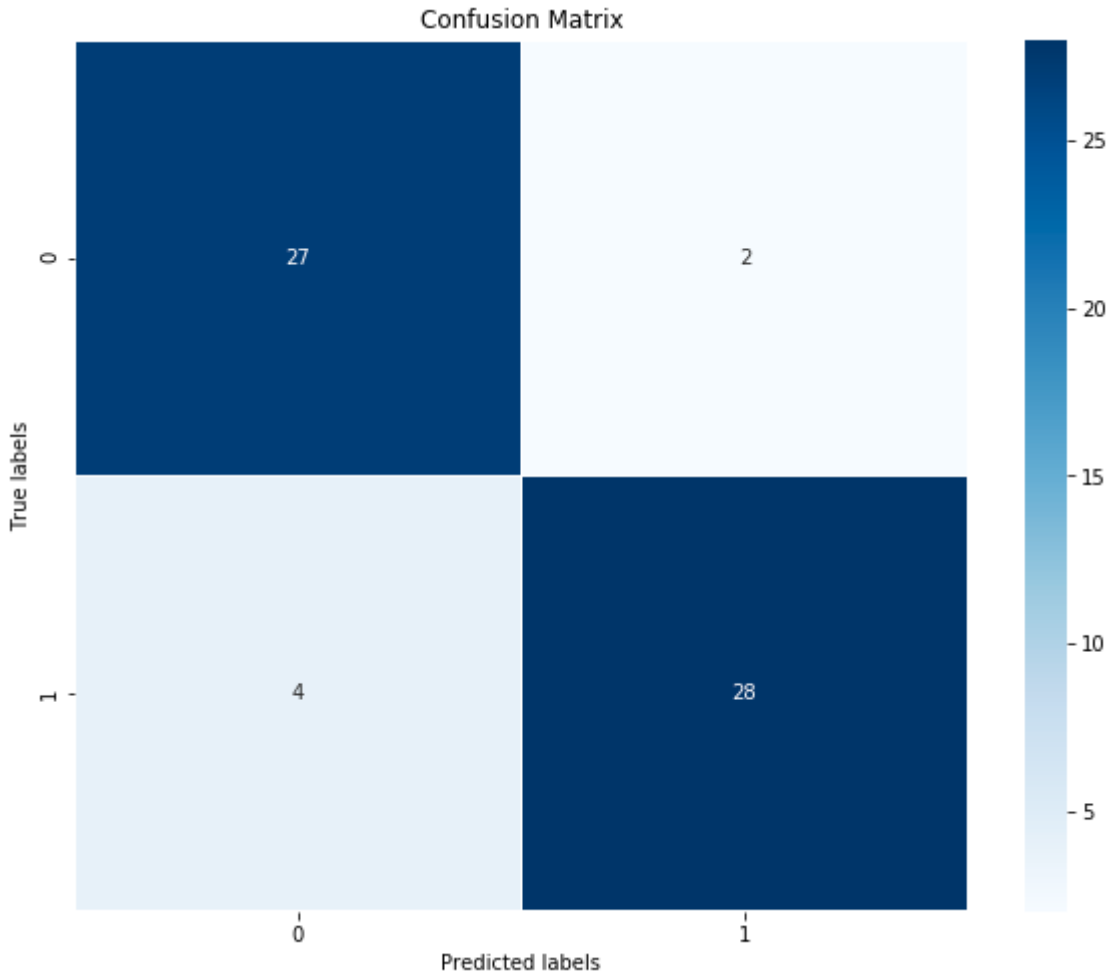
  (# of true positive + # of false negative examples)) * 100

- precision, defined as the proportion:

  precision = (# of true positive examples/

  (# of true positive + # of false positive examples)) * 100

In order to calculate those numbers, a Confusion Matrix was produced for the results, using sklearn.metrics. The seaborn library was also used to generate the heatmap visualization below:



Using the numbers above, those metrics get to:

recall = (28 / (28 + 4)) * 100 = 87.5 %;

precision = (28 / (28 + 2)) * 100 = 93%

Although those results are reasonable, it is important to pursue improvements to the model that can make the recall rate even higher – this is the type of mistake one wants to avoid the most in health care since it implies predicting that sick people are healthy. The precision rate is also good, although not that critical in this case. Those points will be addressed in the Improvement section.

Justification

As mentioned in the previous sections, Logistic Regression was chosen as the final model for this project, providing a 90.16% accuracy on the testing set.

This was a very good result, considering that previous studies made with this dataset obtained accuracy rates between 74.8 and 78.9%. Hence, by directly comparing the magnitude of those numbers, it is possible to see that the results of the project were stronger than the chosen benchmarks.

Therefore, and taking in consideration the whole architecture and strategies adopted throughout the project, which were thoroughly described in the sections above, it is reasonable to say that the solution have adequately solved the problem in hand.

V. Conclusion

Free-Form Visualization

The final step of the project was to try to get some insight about feature importance. This is an important task when performing supervised learning since it allows to determine which features provide the most predictive power. By focusing on the relationship between only a few crucial features and the target label we simplify our understanding of the phenomenon, which is most always a useful thing to do.

This also allows the possibility of doing Feature Selection, by using only a subset of all the available features and simplifying the information required for the model to learn. Although this might reduce the expected time of training and predicting – particularly useful for extremely large datasets -, it usually comes at the cost of performance metrics, and, for this reason, deserves careful evaluation.

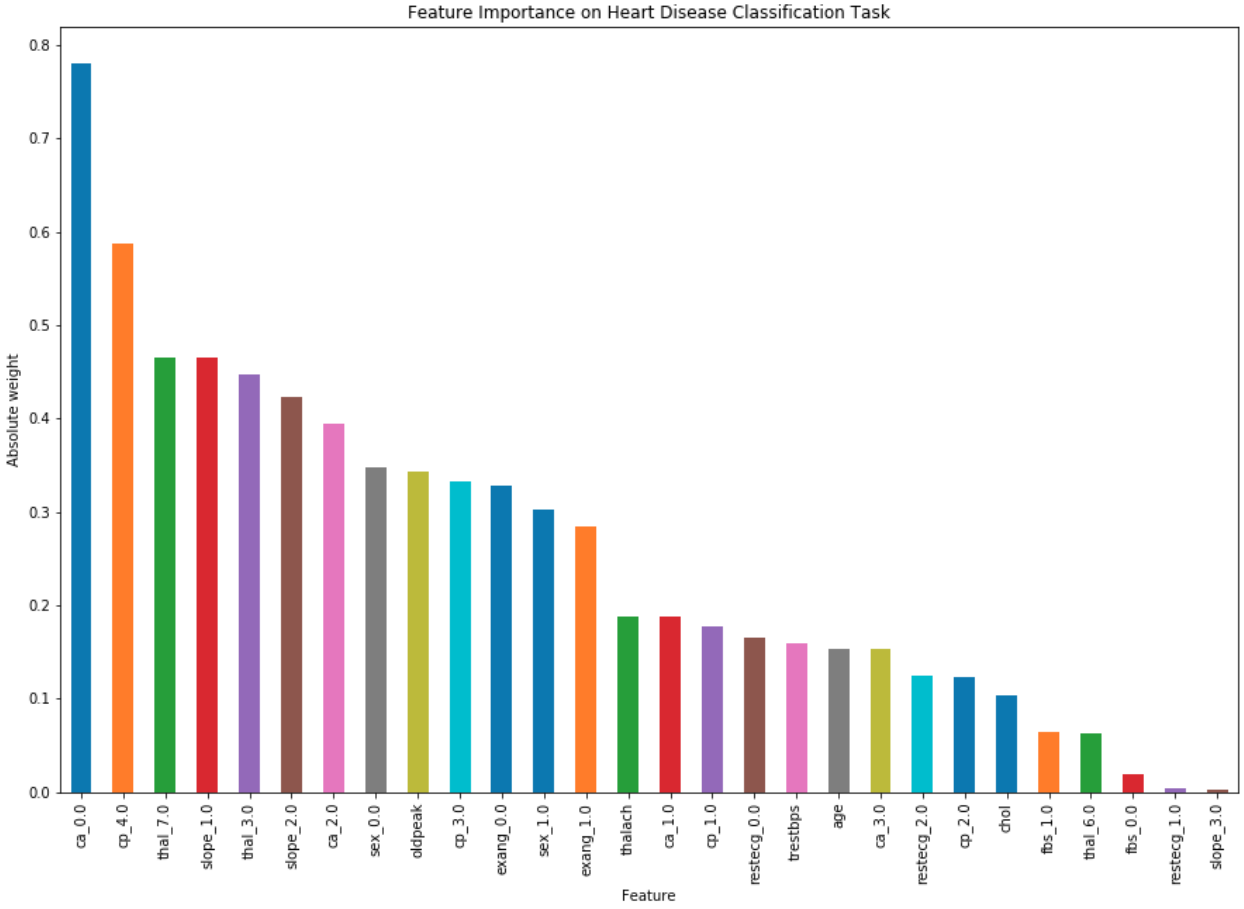
The chosen model, Logistic Regression, does not have a feature_importance_ attribute available for it. Hence, two strategies were adopted and compared.

First, considering that the features were scaled, a ranking of the most important features was created using the absolute weights of the model. The results are shown in the graph by the end of the section, in which is possible to see a group of 13 features with the highest absolute coefficients.

The second approach was to apply the Recursive Feature Elimination (RFE)[6] method, that uses the model accuracy to identify which attributes contribute the most to predicting the target attribute. It can be imported from

sklearn.feature_selection, and allows the choice of the number of features to be selected as top ones.

The RFE method was first applied for 13 features so the results could be compared to the absolute weights graph below:



As a result, 10 of the 13 features considered as most important by the absolute weight method were also considered most important by the RFE method. When trying to choose the top 3 features, the results were a perfect match: both methods agreed on the top 3 selection, that contains the following features:

- ‘ca_0.0’: number of major vessels colored by fluoroscopy;
- ‘cp_4.0’: chest pain type;
- ‘thal_7.0’: thallium stress test.

Reflection

Having a dataset of people with their characteristics and health condition, it was possible to build a model, based on Logistic Regression, that can be used to predictions in new, unseen data. The solution included trying several different algorithms and sets of parameters so that the final model could be the one the brought the best results along the way. The accuracy obtained was satisfactory and exceeded the defined benchmarks.

The project took a lot of research, specially when thinking about which methods and strategies were best suited for the problem and for the mixed type of variables presented as input (both numerical and categorical). In particular, the feature importance section was challenging since there are no built in attributes for that in the method chosen.

Another point that demanded attention was the size of the data set, that is not extremely large and required techniques as regularization and K-fold cross validation to assure that the data could be entirely used, but in a way that would not bias / impact the results.

Improvement

There are two main aspects that can be though of future improvements for this project.

First, the use of the ‘recall’ measure as a metric for the learning process. Although the results obtained can be considered satisfactory, it is important to make the recall an absolute priority for health projects. One immediate challenge is that no benchmarks were found for this metric - only for accuracy -, but maybe the results of this study can be a starting point.

A second improvement would be to evaluate the pros and cons of simplifying the learning process by using feature selection. Although the Heart Disease dataset is not large, not requiring extra time to training and predicting, this could be an issue in the future, if more data is obtained. Working with a smaller set of features also makes it easier to better understand the model and analyze its results.

References

1- https://www.who.int/cardiovascular_diseases/en/

2- https://www.cdc.gov/heartdisease/facts.htm

3- https://archive.ics.uci.edu/ml/datasets/Heart+Disease

4- Dataset source:

> Creators:
> 1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
> 2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
> 3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
> 4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.
> Donor:
> David W. Aha (aha'@'ics.uci.edu) (714) 856-8779

5- https://www.svds.com/learning-imbalanced-classes/

6 - https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/heart-disease.names

7 - https://machinelearningmastery.com/feature-selection-machine-learning-python/