UNIVERSITÀ di VERONA
Dipartimento di INFORMATICA

Analisi di dati per scienze biomediche

SOLVING A BINARY CLASSIFICATION PROBLEM WITH *Machine Learning* TECHNIQUES:

Heart disease classification using Random Forest and Support Vector Machine algorithms

Prof. Lorenza Brusini

Fabio Castellini

PROJECT OBJECTIVES

a) **Implement**, **test** and **compare** two *machine learning* **supervised learning techniques for classification**: *Support Vector Machine* and *Random Forest;*

b) **Tune** the algorithms' **hyperparameters**, perform ***k-fold cross validation***, and extract the performance metrics;

c) Use two **EXAI** (*EXplainable-AI*) methods to further interpet the results;

d) **Compare the performances** of the two machine learning approaches.

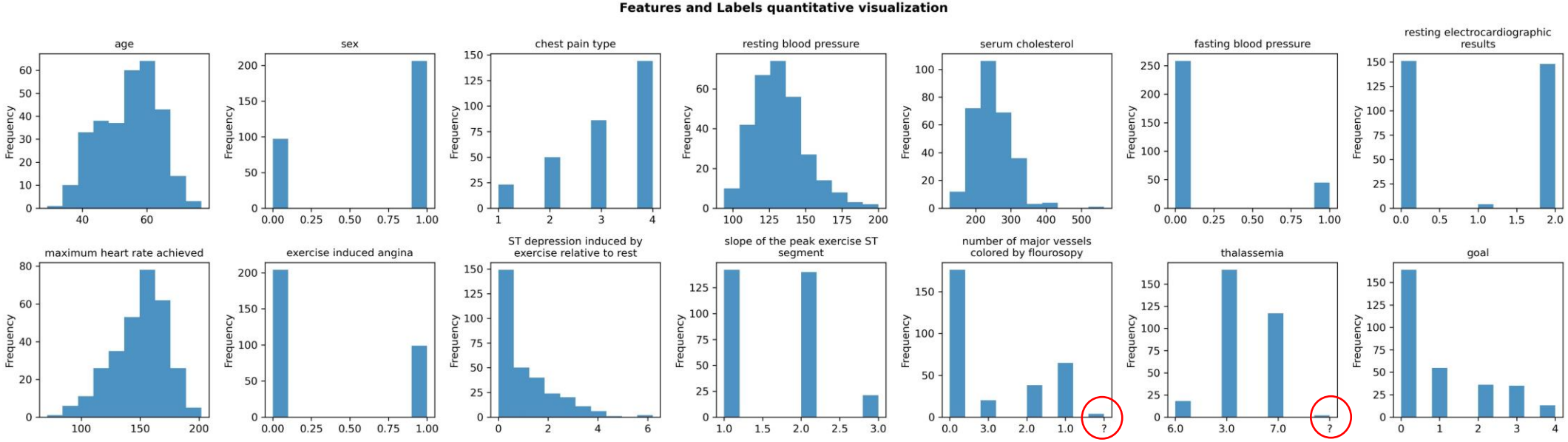
The DATASET: *"Heart Disease"*

- Collection of 303 instances with 13 features: ***age***, ***sex***, ***chest pain type***, ***resting blood pressure***, ***serum cholesterol***, ***fasting blood pressure***, ***resting electrocardiographic results***, ***maximum heart rate achieved***, ***exercise induced angina***, ***ST depression induced by exercise relative to rest***, ***slope of the peak exercise ST segment***, ***number of major vessels colored by flourosopy***, ***thalassemia***.
- The presence of a heart disease is represented by the "goal" column ranging from 0 to 4, that was mapped into booleans (0 = not present; >0 = present).

age	sex	chest pain type	resting blood pressure	serum cholesterol	fasting blood pressure	resting electrocardiographic results	maximum heart rate achieved	exercise induced angina	ST depression induced by exercise relative to rest	slope of the peak exercise ST segment	number of major vessels colored by flourosopy	thalassemia	goal
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
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
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
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
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
...	...	...	...	...	...	...	...	...	...	...	...	...	...

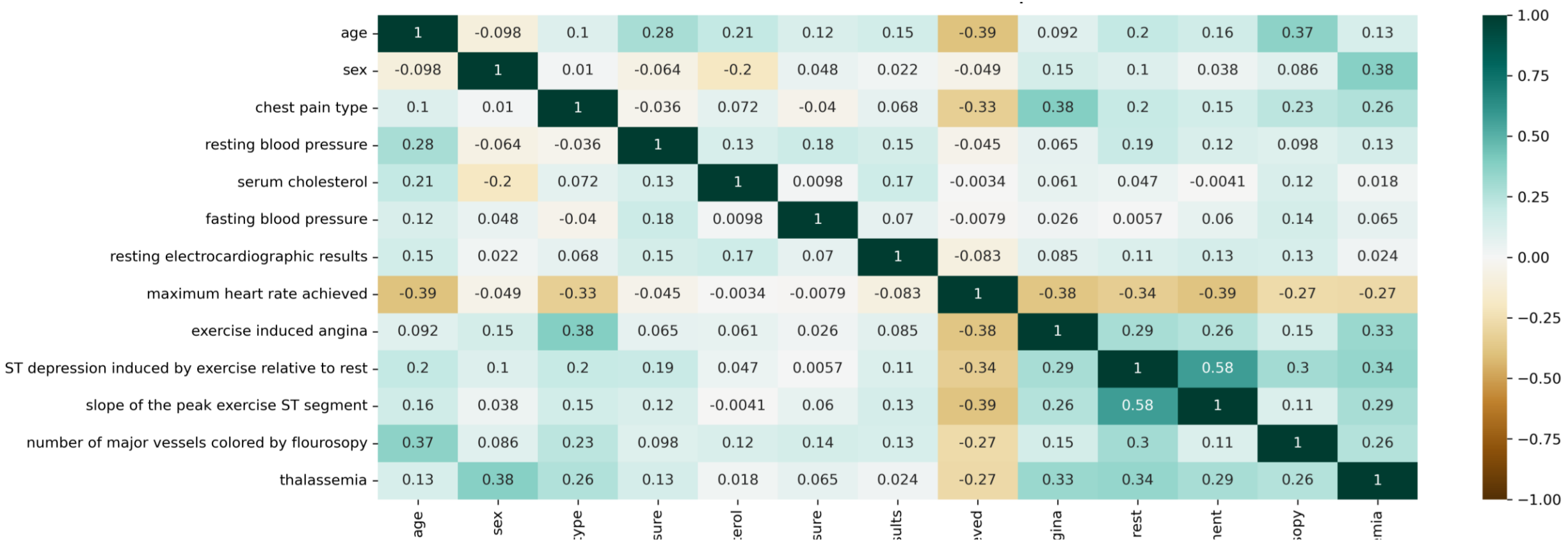
The DATASET: *"Heart Disease"*

- In the following histograms, **features** are visualized to understand their distribution. The classes are well ***balanced*** (*goal* histogram) being ***139*** the subjects suffering from heart disease and ***164*** the healthy ones.
- What stands out is that *not* all features are *equally distributed* (ex: there's not an equal number of male and females). Also, features' scales are different, so normalization (*data scaling*) is required and there are some "?" values to be "cleaned" from the dataset.



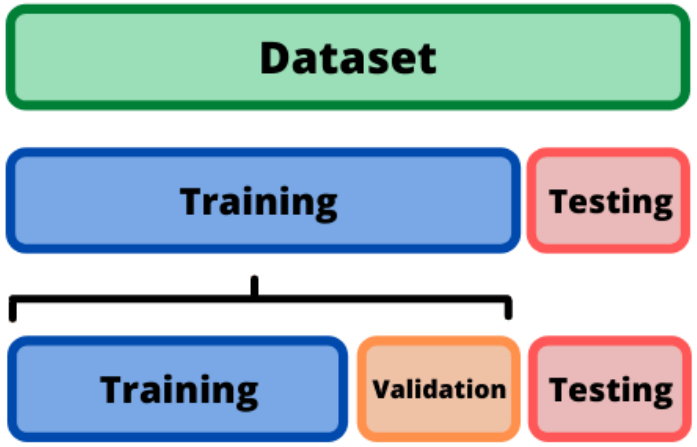
The DATASET: *"Heart Disease"*

In the following figure, a correlation heatmap of the features is shown (for space reasons the bottom legend has been cropped):



Models' TRAINING & TESTING pipeline

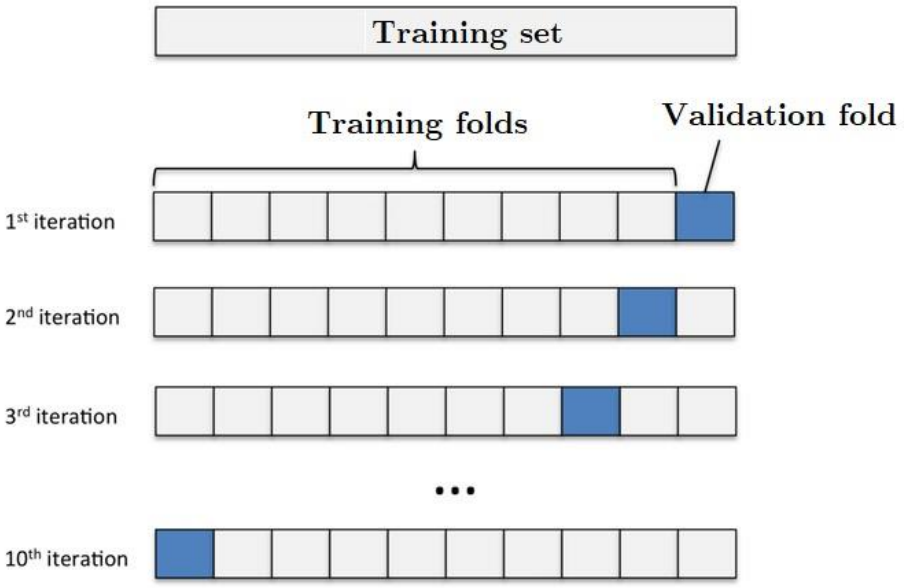
- First of all, the dataset (303 records) is **split**, with the Scikit-Learn function, following the ***80/20 rule*** of thumb. Meaning that 20% of the dataset was kept unknown to the model, until the final testing process. It's worth mentioning that the train/test split was performed with the *"shuffle"* parameter set to *false*, avoiding considerable changes in performances from a run to another.



- For both SVM and RF, **hyperparameters' tuning** is performed through *sklearn.model_selection.GridSearchCV* function. In particular, *C* and *kernel* hyperparamters were tuned for SVM, while *maximum depth*, *features* and *samples* were tuned for RF.
- Once the best models' parameters were found, ***SK-Folds Cross-Validation*** process has been performed.

Models' TRAINING & TESTING pipeline

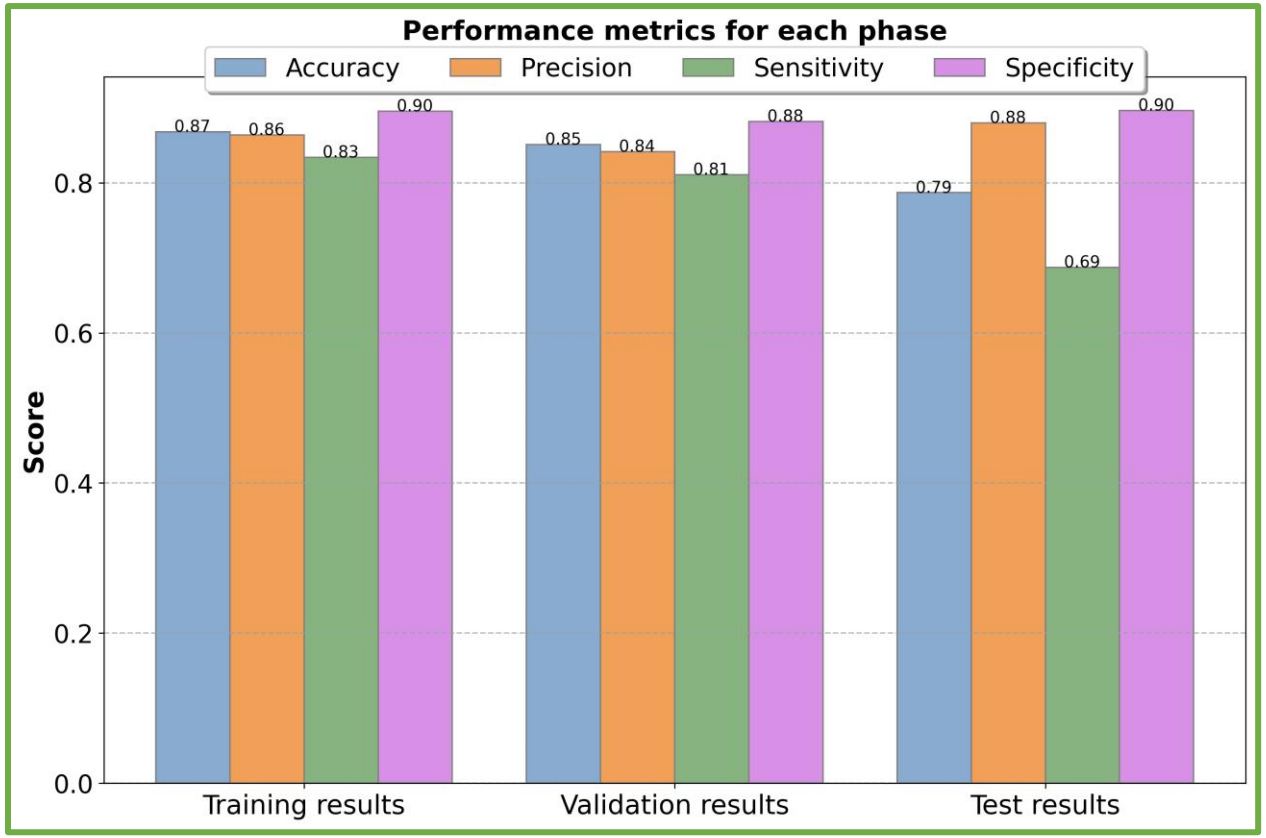
- To evaluate the training process, ***SK-Folds C.V.*** was used (*StratifiedKFold(n_splits=10, shuffle=True)* for both *SVM* and *Random Forest*).
  It consists in splitting the training set into "n_splits" portions, train on "n_splits-1" samples and evaluate the model on the remaining one. Repeating the process over all folds, we retrieve *training* and *validation* performances and optimize the hyperparameters' selection.

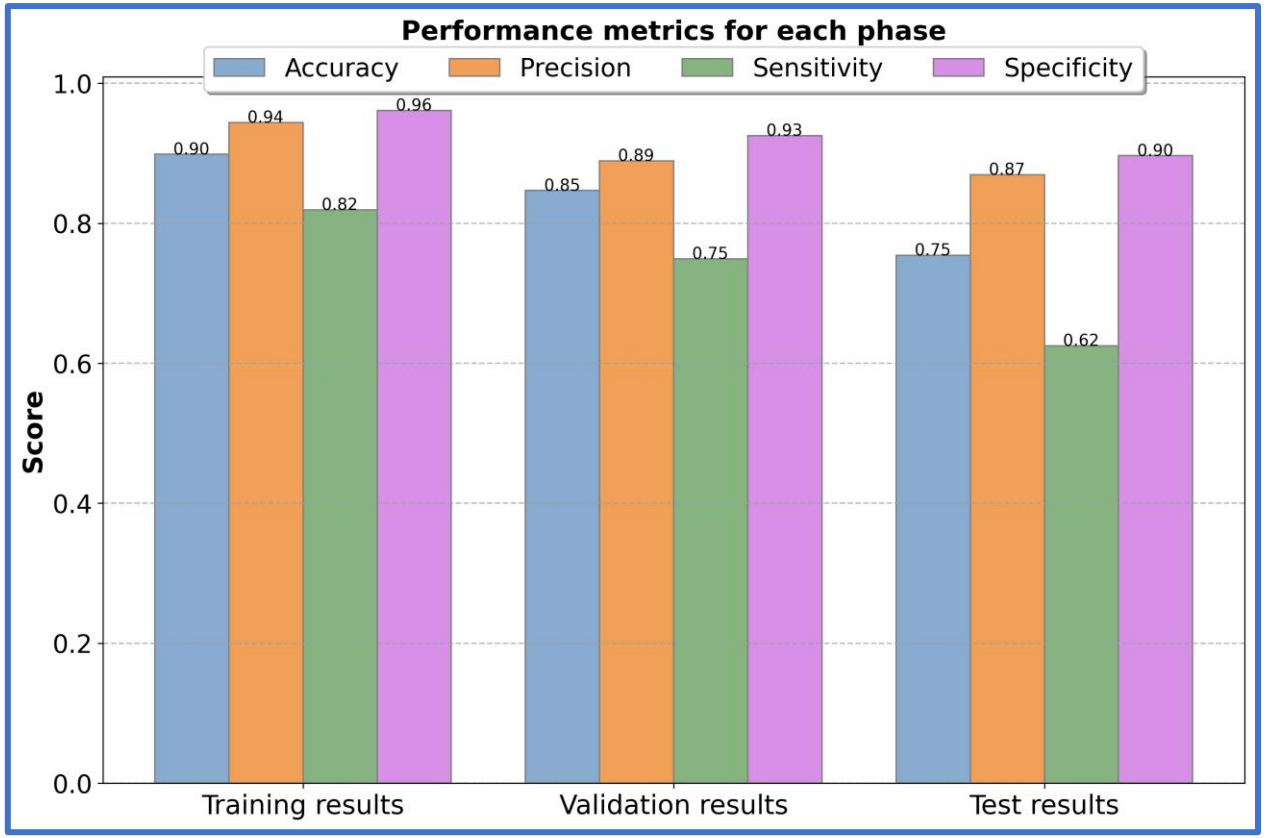


- Finally, the fitted models were ***tested*** on the never seen test portion and several performance metrics were retrieved: train-validation-test accuracy, precision, recall, specificity; Receiver Operating Curves, classification reports and confusion matrices.

Support Vector Machine vs Random Forest performances

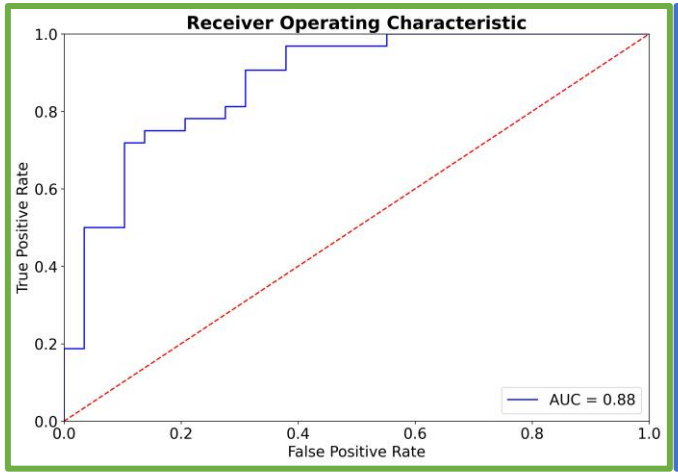
*The following histograms show accuracy, precision, sensitivity (recall) and specificity results for **SVM** (left) and **RF** (right):*

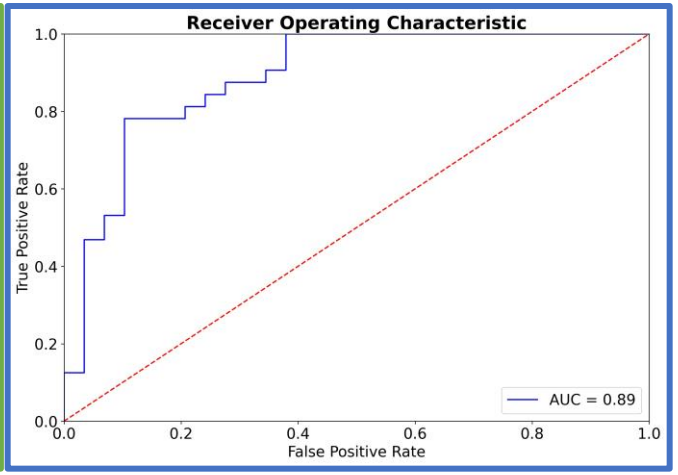


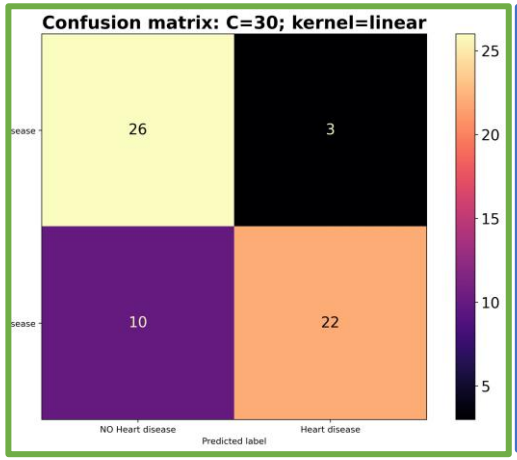


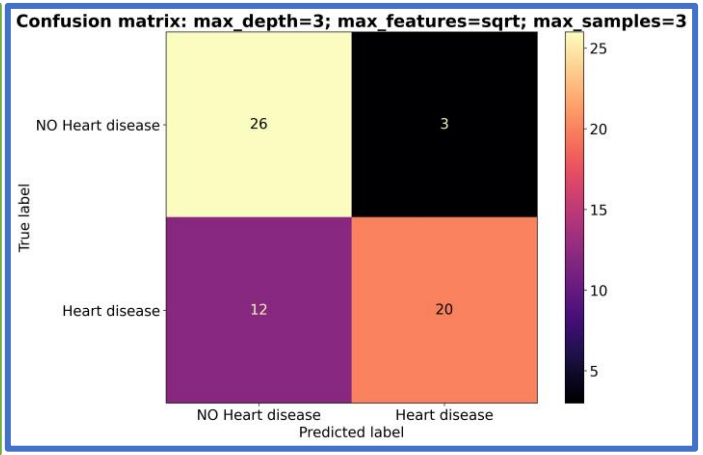
Support Vector Machine vs Random Forest performances

*The following figures show **ROC curve** (sensitivity(1-specificity)), **confusion matrix** and **classification report** for **SVM** (left) and **RF** (right):*









	precision	recall	f1-score	support
No heart disease	0.72	0.90	0.80	29
heart disease	0.88	0.69	0.77	32
accuracy			0.79	61
macro avg	0.80	0.79	0.79	61
weighted avg	0.80	0.79	0.79	61

	precision	recall	f1-score	support
No heart disease	0.68	0.90	0.78	29
heart disease	0.87	0.62	0.73	32
accuracy			0.75	61
macro avg	0.78	0.76	0.75	61
weighted avg	0.78	0.75	0.75	61

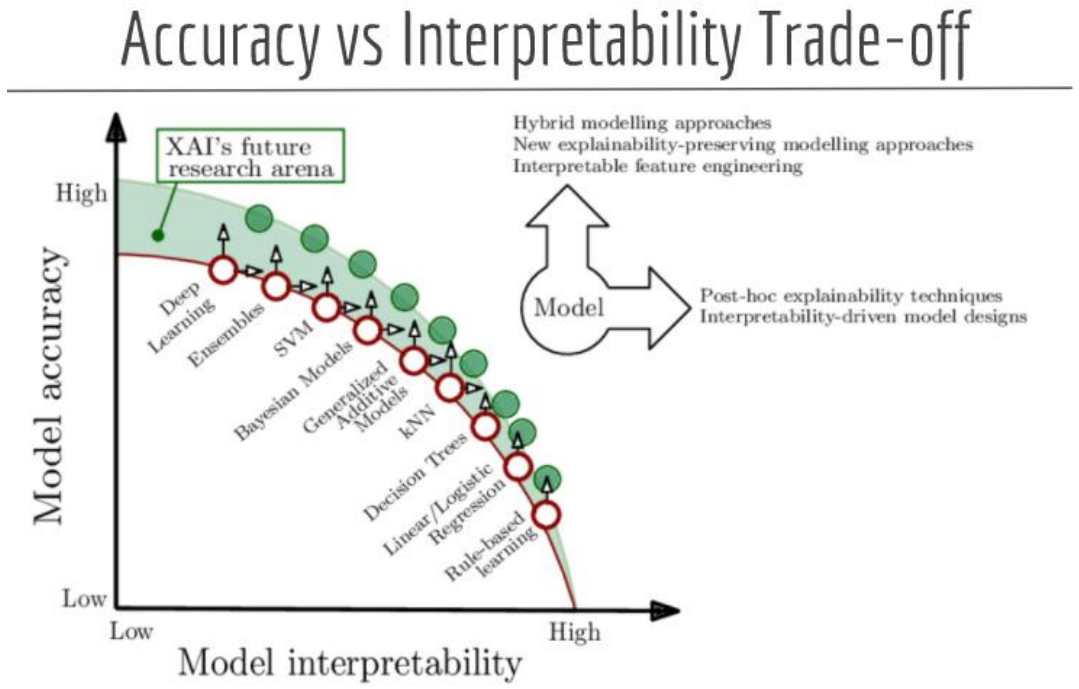
Support Vector Machine vs Random Forest performances

Comments about the previous results

- Performances, in terms of accuracy and precision are in line with what shown on the dataset's page (*SVM*: 0,55-0,76 accuracy & 0,58-0,82 precision; *RF*: 0,71-0,88 accuracy & 0,73-0,90 precision).
- Between the two machine learning algorithms there aren't huge differences in terms of performances. Looking at the considered metrics, *SVM* seems slightly better, but as previously mentioned, ***the initial train/test split strongly affects overall results***.
- Looking at the classification reports, in both cases there's a high precision / low recall pattern for "positive heart disease" and vice-versa for negatively classifying the samples. I'd say that this is a *good behavior*, in fact the model is pretty sure when detects heart diseases, even if it doesn't always detect them. On the other hand, negative heart diseases are well detectable by the model but can sometimes be misclassified as positive heart diseases.

EXAI for *SVM* & *RF*

- EXAI stands for **EX**plainable **A**rtificial **I**ntelligence and it's a research field aiming at decoding the internal logic of a Machine Learning algorithm. It's useful to understand *how* and *why* a certain prediction or observation is made.



In particular, the following techniques were exploited:

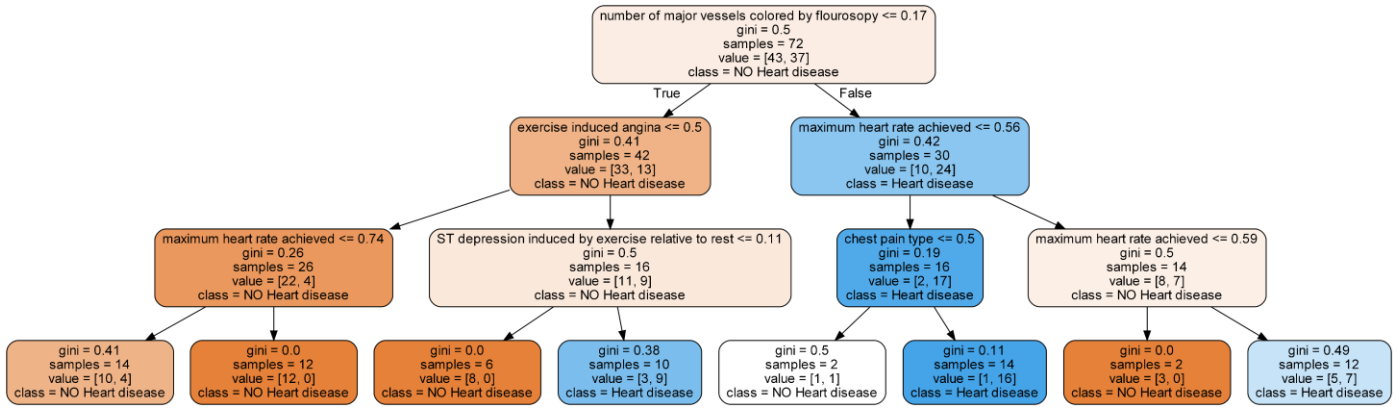
- ***Permutation feature importance***:* run the trained model on several permutations of the input and measure the consequently change in predictions (large change → important feature);
- ***Local Interpretable Model-agnostic Explanations (LIME)***:* perturbation based; the idea is to start from a test-sample and create a new set of samples. On this set, a simple interpretable model can be fitted, to better understand the feature importance (ex: looking at the weights). Computing the average feature-importance scores, a ***global*** explanation can be retrieved.

**Post-hoc, model agnostic*

EXAI for *SVM* & *RF*

- ***SHapely Additive explanation (SHAP)**:* compute the *SHAP values* for each feature (from game theory). These values explain features' importance on the model's predictions. The sum (linear combination) of all SHAP values explain the ***difference*** between the *actual prediction* and the *average prediction of the population*.
  By averaging the SHAP values on all observations, a ***global feature importance*** can be obtained.

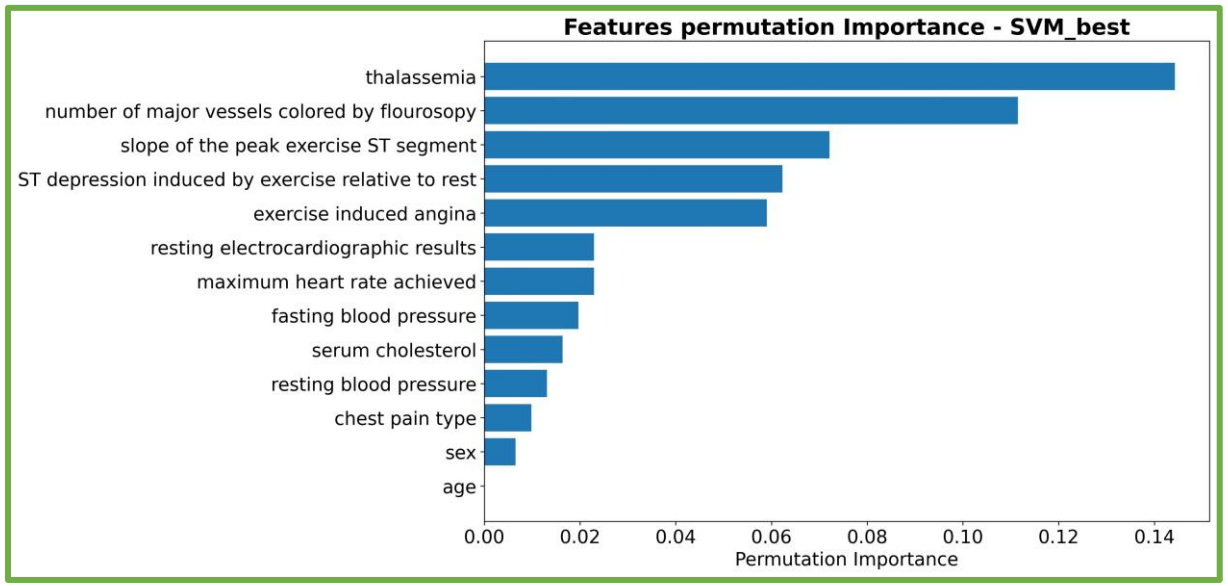
- "Qualitative" representation of a ***decision tree*** extracted from the *Random Forest* tuned and trained model, according to *https://towardsdatascience.com/how-to-visualize-a-decision-tree-from-a-random-forest-in-python-using-scikit-learn-38ad2d75f21c*. A random forest is an ensemble of decision trees, so only one of them (out of the default 100) is considered to generate the image. It can't be an exhaustive representation due to the differences between trees inside the forest, but can be used to get an idea of the decision process.

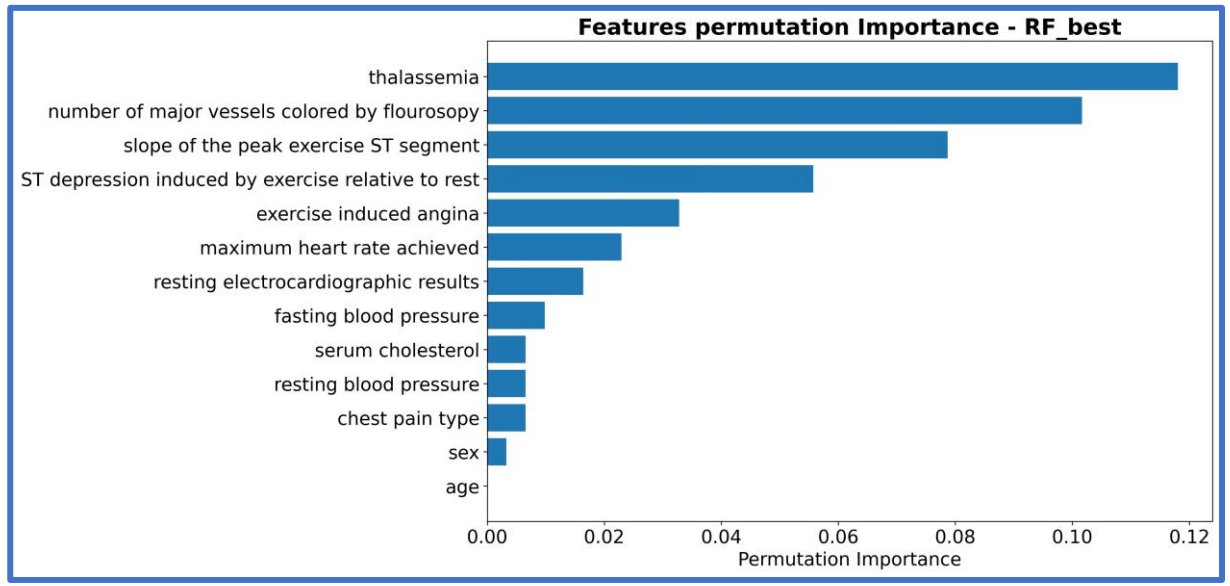


**Post-hoc, model agnostic*

EXAI for *SVM* & *RF*

Permutation feature importance

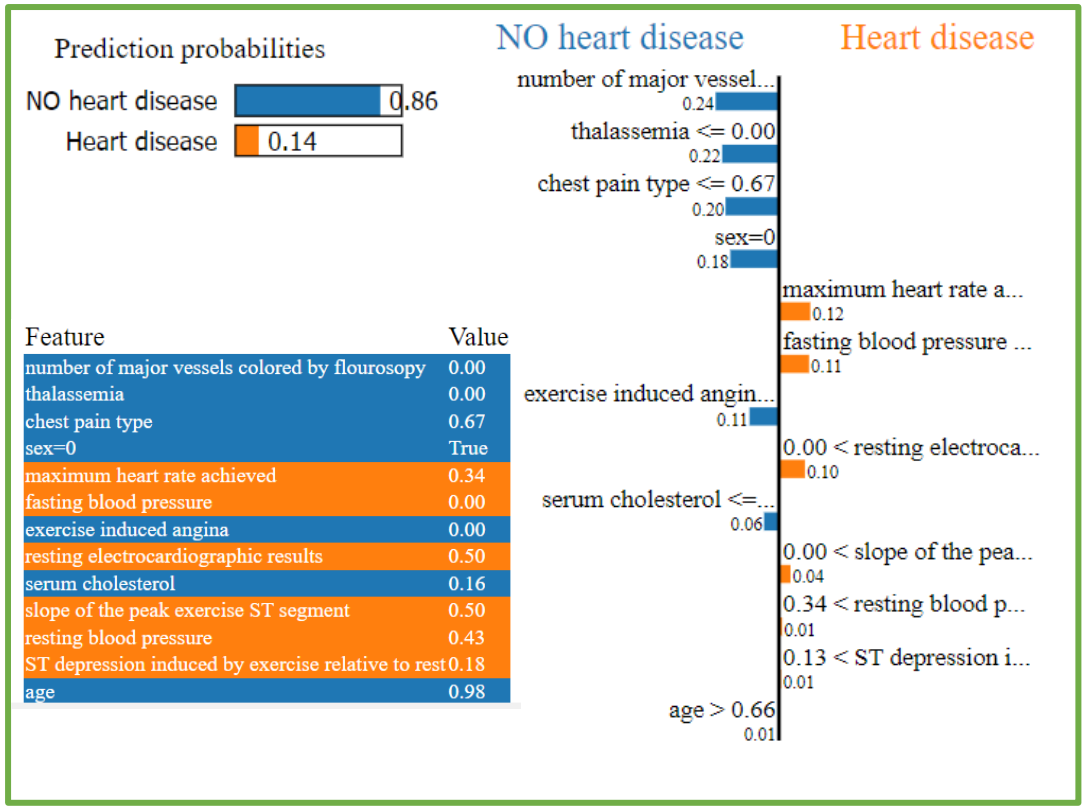


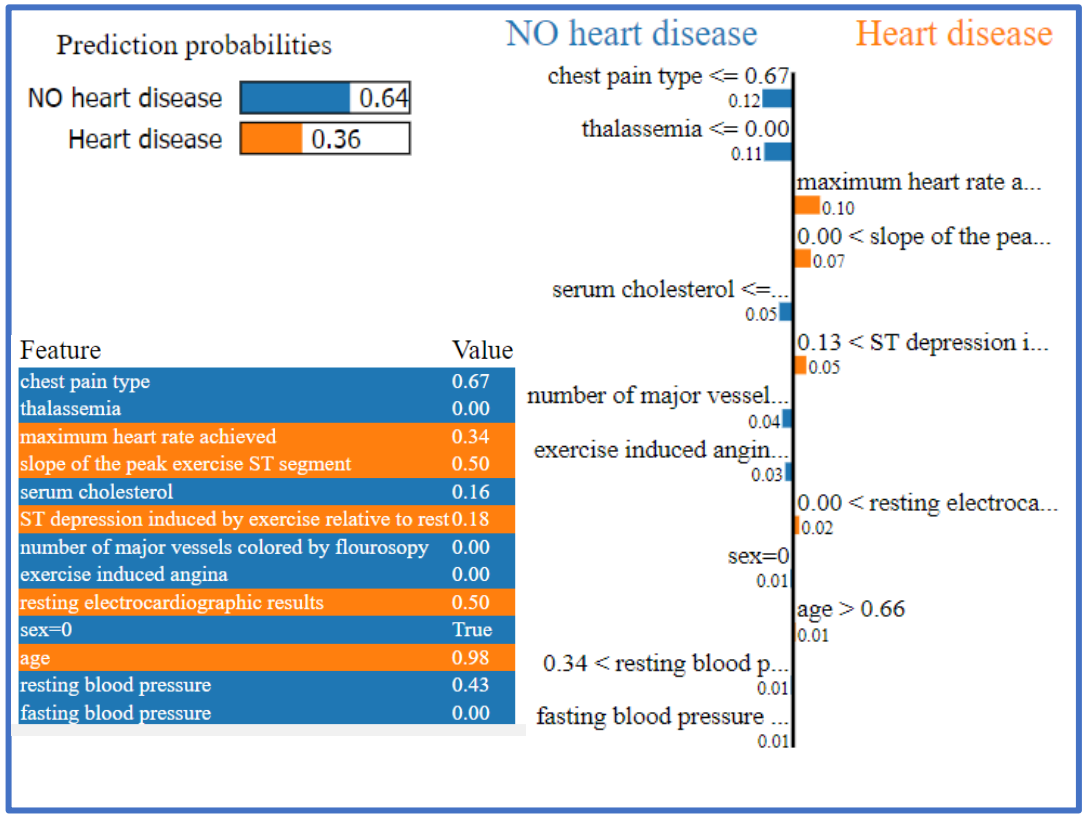


The above bar plots show feature importances computed by permutation.
In this case, the two models have a similar result.

EXAI for *SVM* & *RF*

LIME explanations on a *FALSE* instance

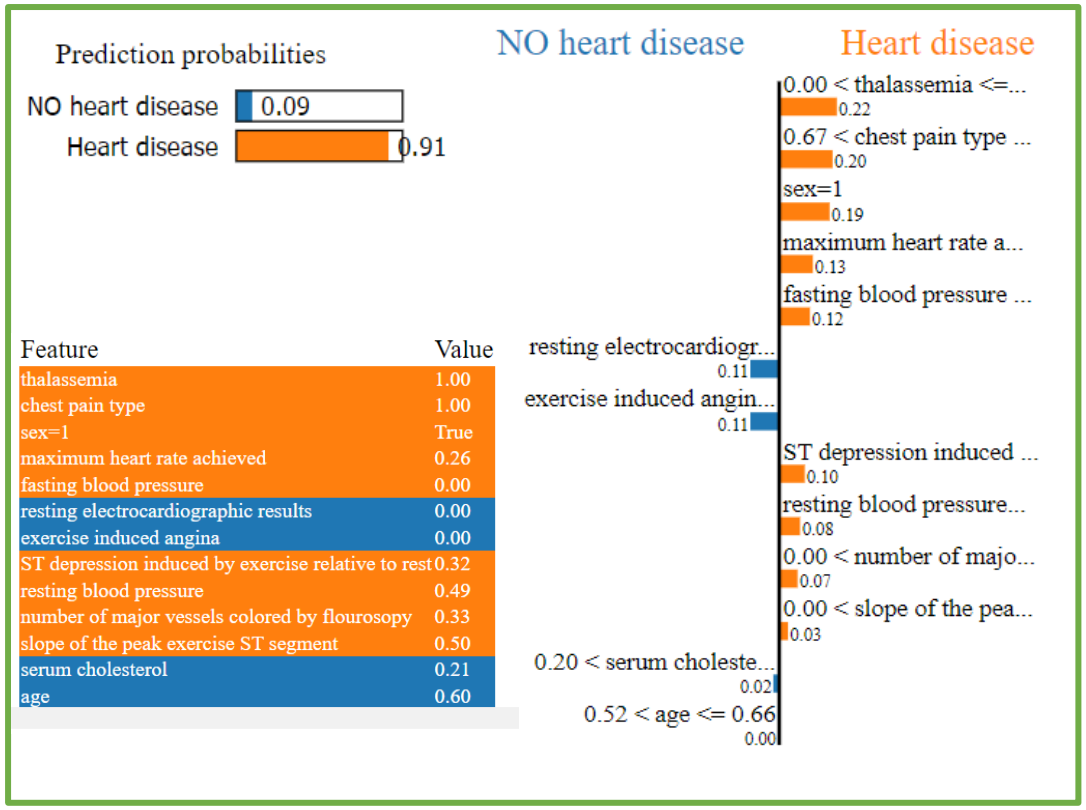


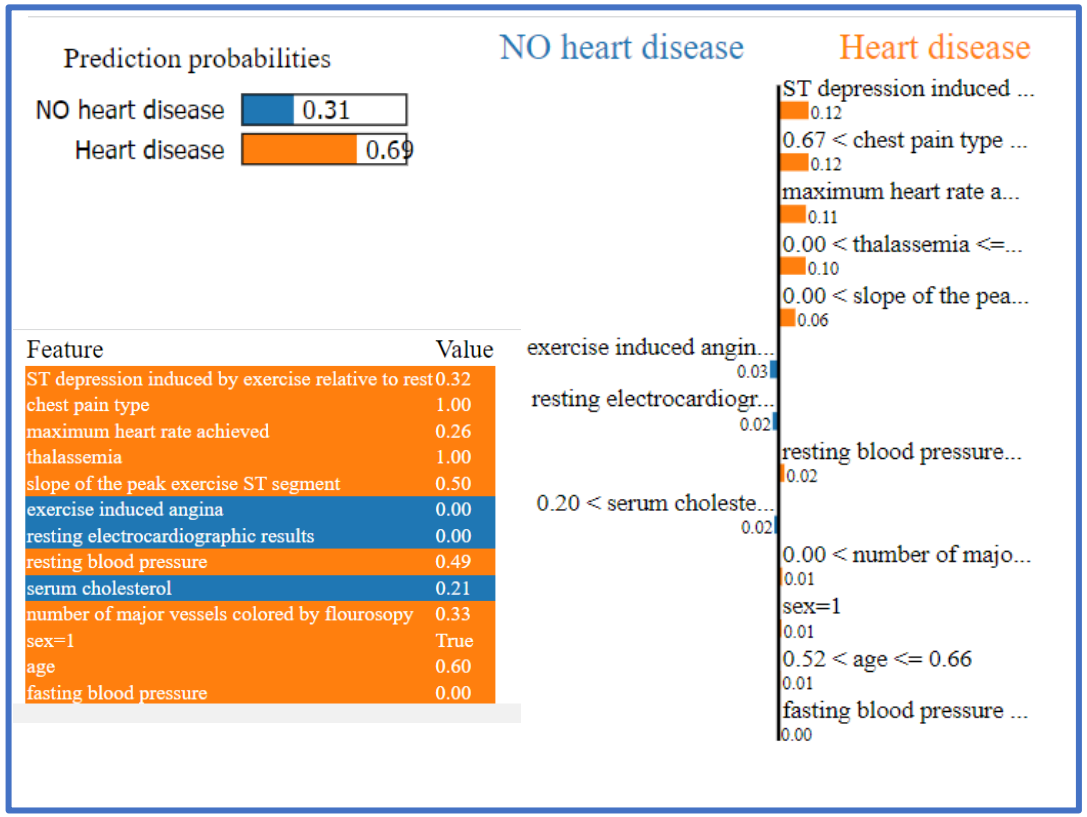


The above figures show the linear regressors' weights that were used to explain a specific FALSE predicition. All features were used to make the explanation.

EXAI for *SVM* & *RF*

LIME explanations on a *TRUE* instance

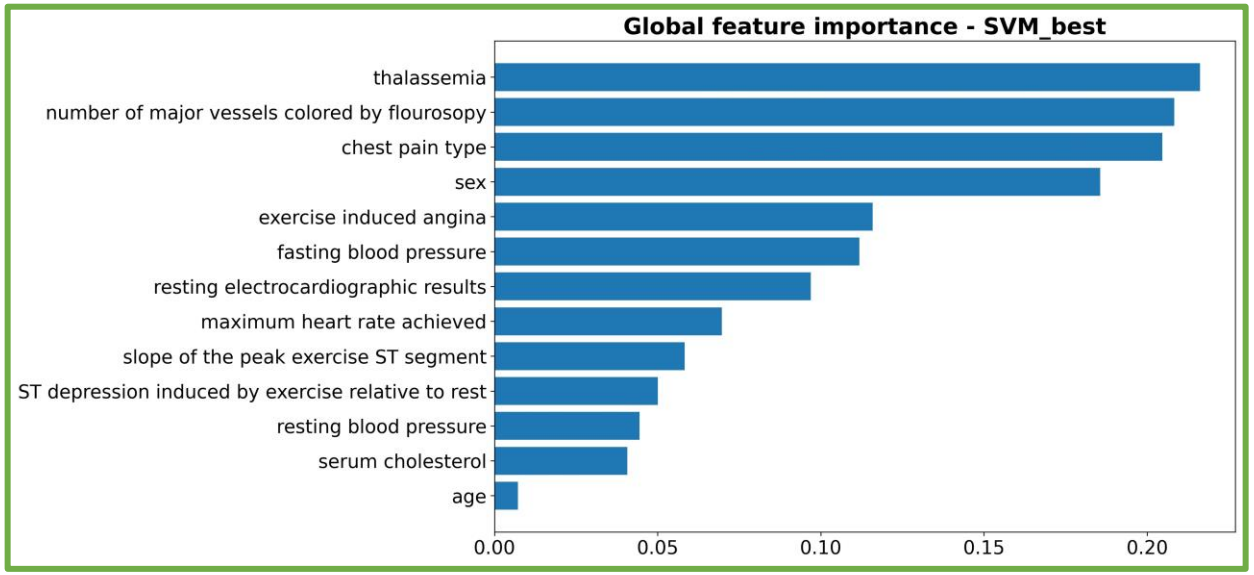


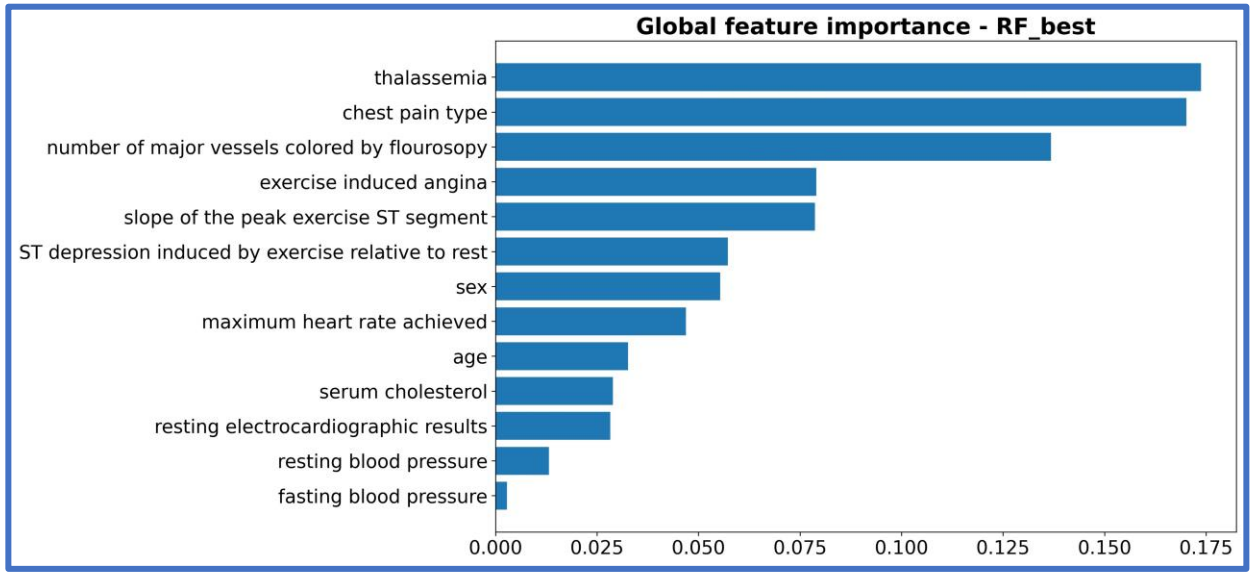


The above figures show the linear regressors' weights that were used to explain a specific TRUE predicition. All features were used to make the explanation.

EXAI for *SVM* & *RF*

LIME global feature importance (cycling through all instances)

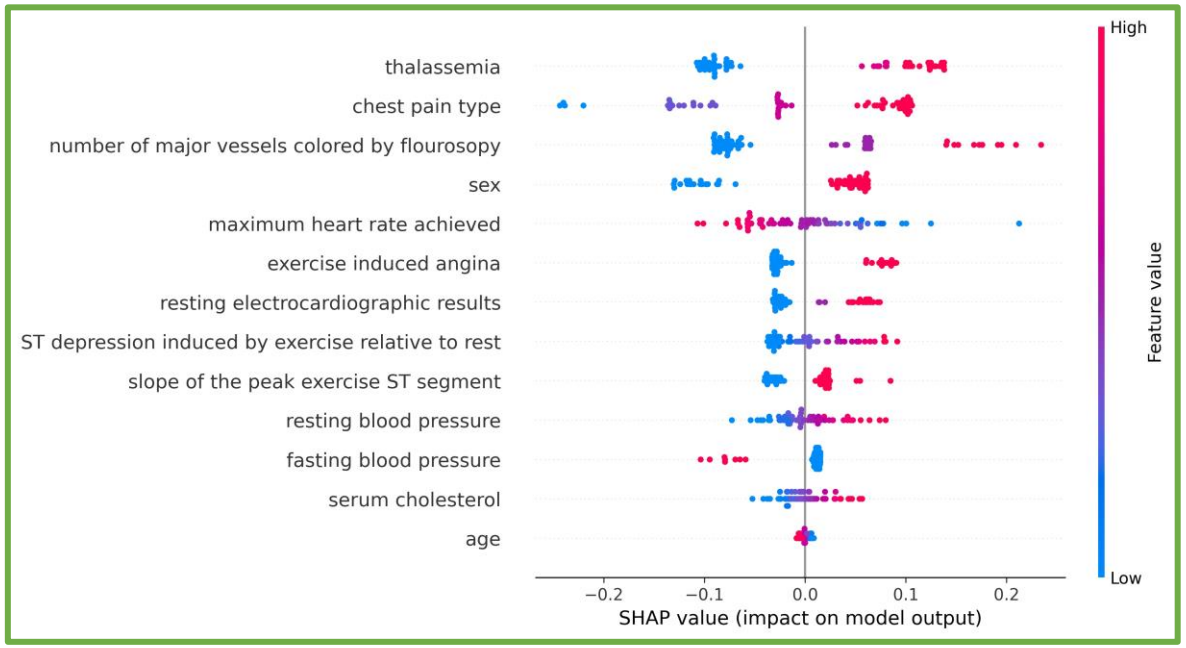


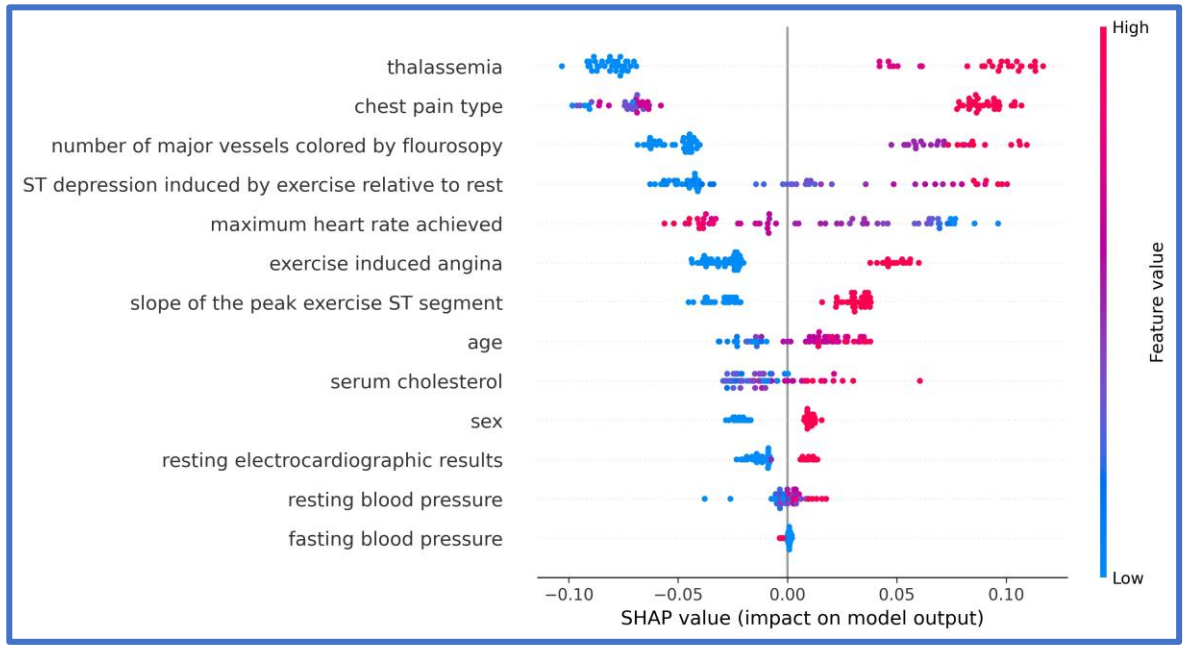


*The above bar plots show the **global** feature importance on the models' outputs, computed by averaging the local feature importances of all test instances.*

EXAI for *SVM* & *RF*

SHAP values for each feature (positive class: *heart disease*)

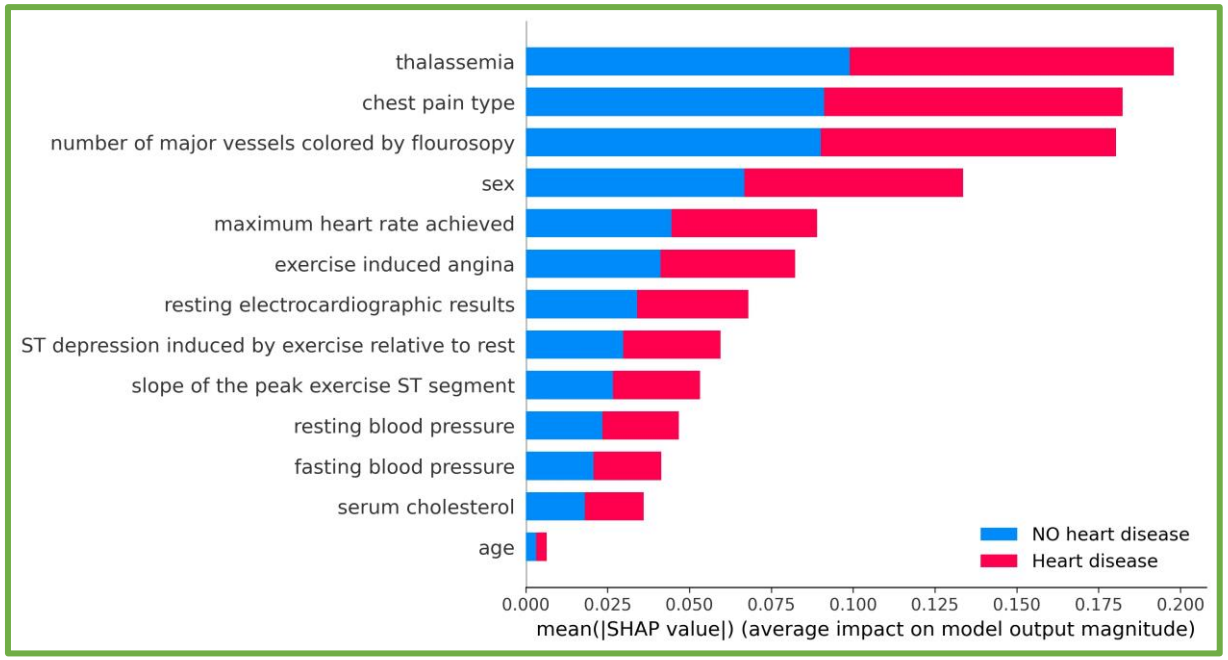


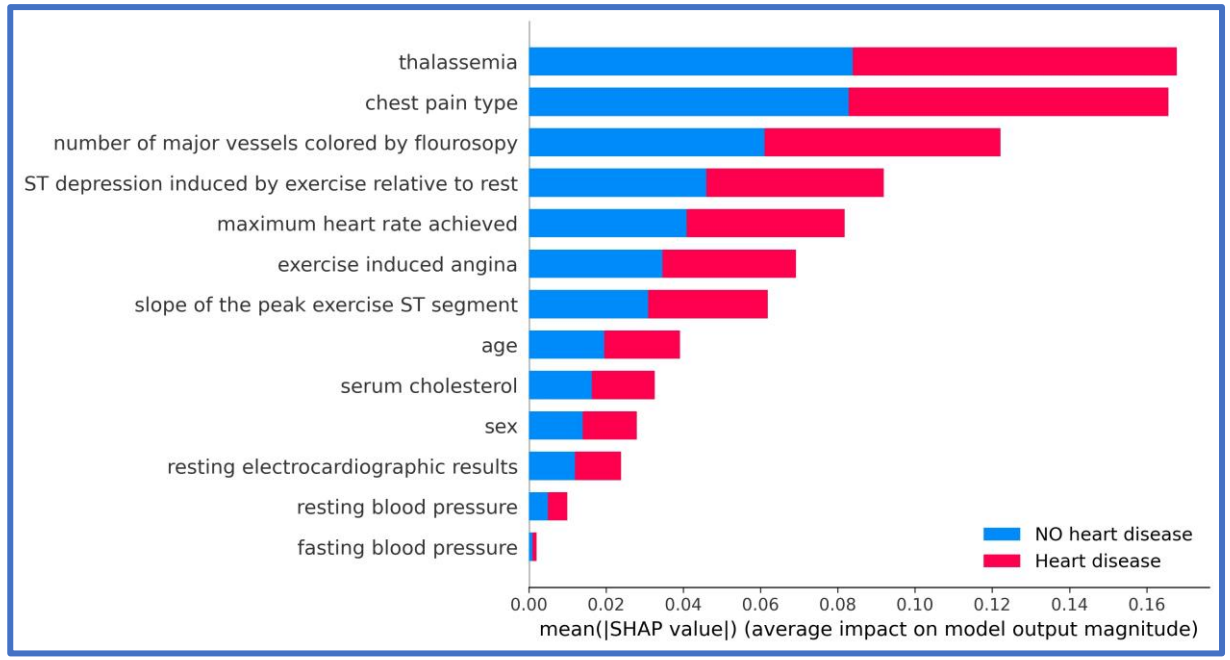


Noticing the x-scale difference, the 13 features bring a ***similar impact*** *on the two models. However, features like 'sex' and 'resting electrocardiographic results' are much more important for SVM than RF, while the opposite can be said for 'ST depression induced [..]' and 'age'.*

EXAI for *SVM* & *RF*

SHAP values for each feature

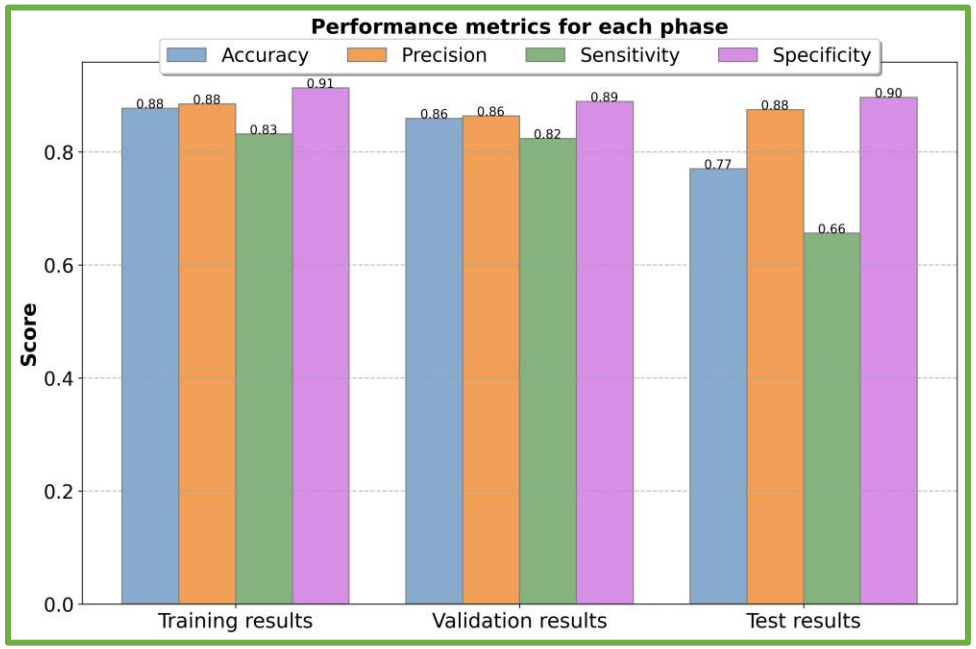


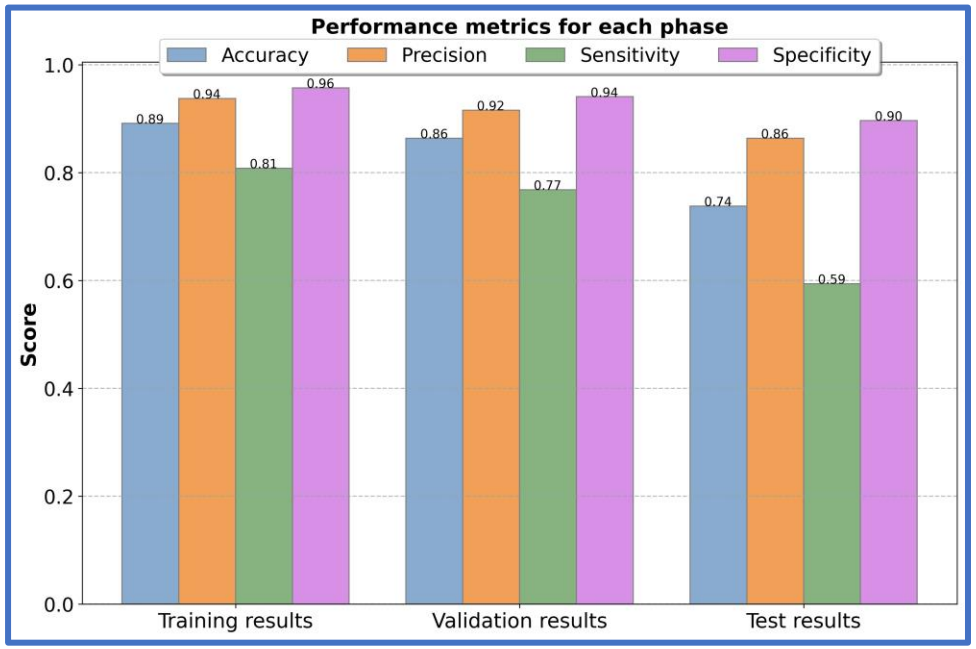


*The above bar plots show the **global** feature importance on the models' outputs, computed by averaging the previous SHAP values. Results are similar, but not identical, to the ones obtained with LIME.*

Some tests on feature importance

- According to the obtained results, *thalassemia* is consistently the most important feature. Several experiments were performed (on the same 80/20 split), removing highly correlated features (*maximum heart rate achieved*) and **removing '*age*' and '*sex*'** that can sometimes be irrelevant (confirmed by some of the EXAI results).

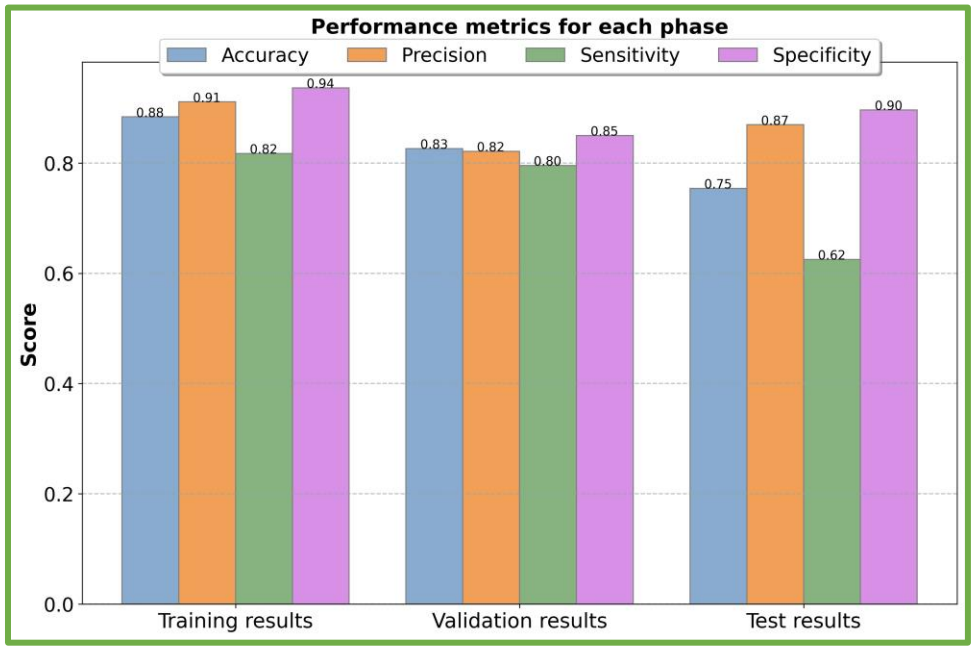


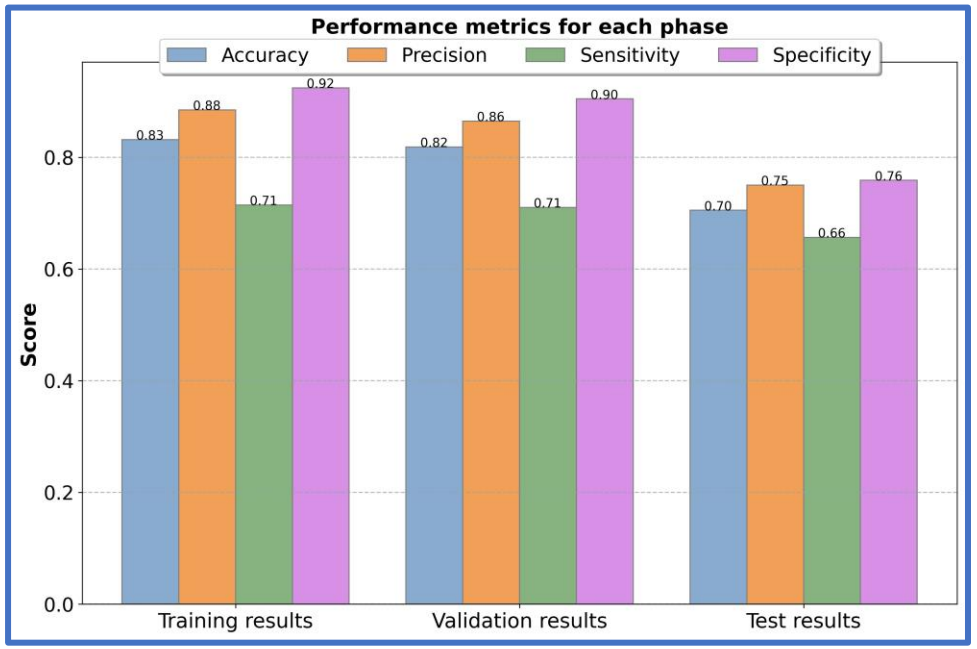


*Slightly worst results (13 features: 79% - 75% acc), removing 'age' and 'sex' features (**SVM** on the left; **RF** on the right)*

Some tests on feature importance

- The following results were obtained **removing '*thalassemia*'**, the most important feature. Random Forest and SVM both show a drop in terms of performances, but not in a so significant manner. What's worth mentioning is that tests are made on just 61 samples and being a ***binary classification problem***, 50% would be the "theoretical random accuracy".

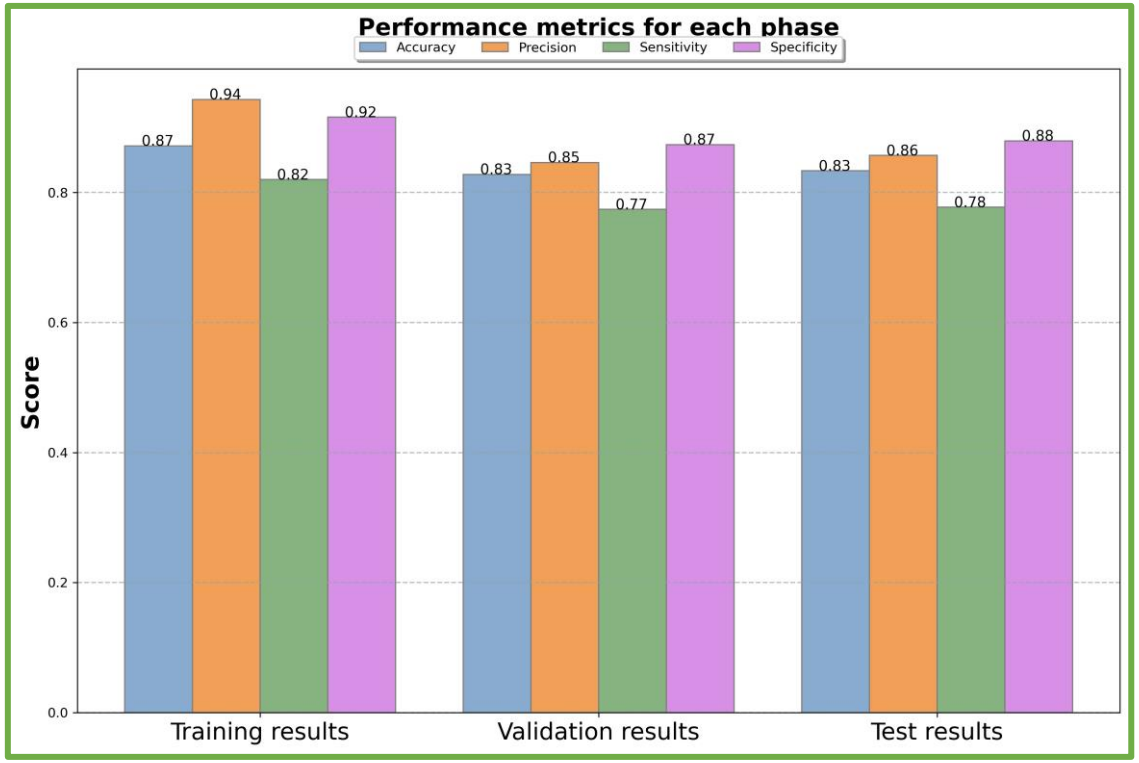


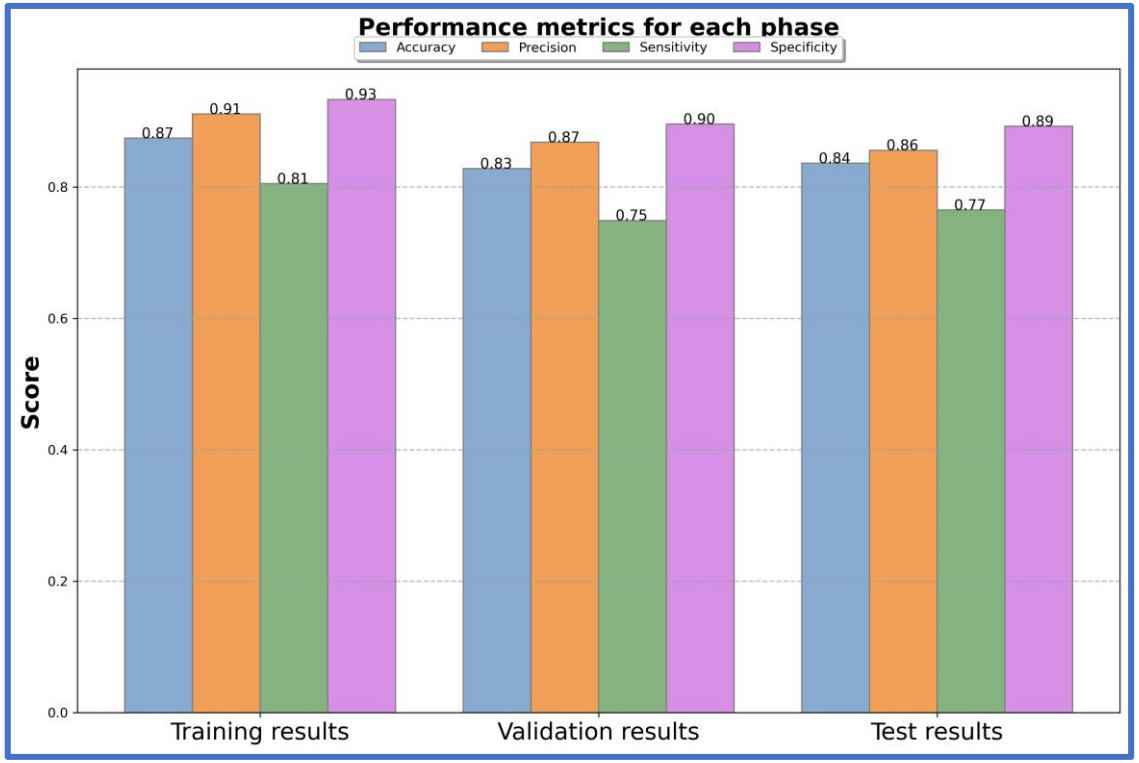


*Slightly worst results (13 features: 79% - 75% acc), removing 'thalassemia' feature (**SVM** on the left; **RF** on the right)*

Additional testing through *Averaged Holdout* (20 times)

- To retrieve more realistic performances on the dataset, that can be independent on the initial 80/20 split, *averaged holdout* technique was used. Accuracy, precision, sensitivity and specificity were averaged over the ***20 iterations*** of both ***SVM*** and ***Random Forest***.





Conclusions

- In this project a ***binary classification problem*** was solved through 2 popular machine learning techniques.
- Results and performances of ***Random Forest*** and ***Support Vector Machine*** were similar, even though the algorithms have a different approach. They have been compared using traditional machine learning evaluation metrics and several EXAI methods.
- Further developments of the project could be to test the trained model on a benchmark external dataset, to evaluate their goodness in a more challenging scenario.