DESIGN:

BUILDING A QUESTIONNAIRE / SURVEY TO UNDERSTAND WHAT IS USEFUL IN THE PLATFORM:

Although the CliniColab currently solves some of the issues with the way Machine Learning is applied to the medical field, others still need to be addressed.

The method used to identify and correct these issues is two-pronged. On one side, an investigation of the state-of-the art tools as well as an in-depth investigation of the root of the aforementioned problems was conducted. This investigation relies heavily on the published papers. This has allowed to have a general view of the options available. Then, a decision is made concerning which tools are best suited to solving the problems we have chosen, and among these which can be implemented and used in an easy enough manner.

The main lessons learned from this approach were that the tool needs to bring explainability to any type of model (model-agnostic), for both categorical, numerical, and mixed data. Moreover, explainability should not be centered on outliers only, or on the general performance or the model. Rather, it should aim to provide granular as well as overall explanations of the predictions and why they were made. Therefore the tool will contain a feature importance computation as well as specific prediction explanations. For the latter, both LIME and SHAP have been considered. Both of these methods have their strengths and weaknesses. However one major issue with LIME is that the adjustment of the kernel-width parameter is tricky and can lead to approximations that are off. This means that LIME will consider a region that is too small to be able to generalize, or too large and will miss out on the specificities of the boundary. However the computation time is advantageous and there is benefit to having more than one method to analyzing a model outcome. SHAP is very robust and will be the default method.

On the other side, the approach revolved around asking the end-users, clinicians, their opinion regarding the platform. To this end, a survey was written which aimed at letting the person answering be as little constrained or guided as possible, so as to get an authentic feedback. The surveys were sent after the clinicians had an opportunity to observe the platform and to discuss its main points and functionalities.

One of the points that was brought forward the most often were that there needed to be ass many functionality tools as possible while keeping the overall platform complexity low. The solution to this would be to have a default setting to the tool which only enables some of the functionalities, and the option for more experienced users to dive deeper into result analysis with more functionalities.

Another point brought forward was that there needs to be some type of information on how to use the platform and what the explainability tools aim to do. To this end, a use case detailing all of the platform functionalities has been created, complete with some explanations on why some results make sense, using a medical dataset. Another complementary part to this information guide was to create both a tutorial on how to use the platform as well as a tutorial on how machine learning can be used.

There will be other functionalities that are based on specific use cases and datasets. The main one is the construction of a severity score of patients in order for a clinician to be able to select only the most at-risk patients from a group.

Once this severity score has been computed, some graphs can be made. Each patient / row from the dataset can be allocated to a severity group. Then, analysis of results can be done per-group. Techniques such as feature importance can be run again only on a specific group to better understand how the model classified a certain at-risk sub-population.

BUILDING A COMPREHENSIVE MACHINE LEARNING MODEL INTERPRETATION TOOL:

In order to build an interpretation tool that is useful to the end-users, one has to consider the nature of the dataset that will be used as well as the intent of the user, and their level of knowledge regarding machine learning.

The datasets used on the platform up to now pertain to data collection within clinics in Tanzania. There are many different features, and not all are harmonized during data collection. One of the possible labels to detect is the patients being malaria positive or negative. There are many missing values in the dataset and only a subset of features make sense to be included in the detection of malaria. One example is that of the geographical location of the clinic visit. In theory, this information would be greatly helpful to detect cases of disease spreading. However, the platform for now does not seek to detect evolution of diseases over time. Moreover, the number of missing values is too high to be able to recover a sub-dataset that is large enough for network training.

If all features are included in the model training, the results might become confusing (more than a hundred features with a good number of them containing more than acceptable levels of missing values. Therefore the platform has already implemented a mechanism to take out some of the features without even trying to include them in the model training. To take this one step further, the platform can also apply feature reduction techniques such as applying Lasso or Boruta. Lasso uses an L1 penalty factor on feature coefficients during the training process. Cross-validation can be used to choose this parameter in an optimized fashion. Since the platform is not using optimization for now, the best method would be to use Boruta. Boruta effectively tests a feature column’s effect to the overall model result against the same column with values mixed up within. In the latter case the model is trained using a feature column that is “randomized”, and the model results will show a decrease in metric scores if that feature was actually important to the model prediction.

The dataset contains both numerical and categorical values (both binary and multi-categorical). It is expected that all binary values are marked with a zero value if to represent the absence of the feature, and 1 for the presence of the feature.

After having discussed the dataset specificities, one of the conclusions to be drawn regarding the manner to investigate the model explanations. For the end-user to analyze results, many features need to be discarded first. Missing value thresholds and general knowledge about the problem at hand should be the two leading criteria to make this cut.

it is now important to discuss the machine learning problem at hand. The main task appears to be disease diagnostics. This is a classification type problem. The advantage of disease diagnostic is that there are diseases that can be diagnosed with high efficacy point-of-care tests. Therefore, the dataset’s labels are accurate at an acceptable threshold, which is not always the case in medical datasets. An important point to mention is that when dealing with not just binary classification, but multi-class labeling, new problems might arise. For a binary outcome; does a patient have a disease or not, the data is either pushing towards the presence of a disease or towards the absence of it. Different diseases have specificities, and symptoms that make them identifiable. However, many have symptoms in common. If a model has to choose between five different diseases, with the possibility of a patient potentially having multiple diseases, the model training becomes complicated. The mathematical logic behind some models should make it possible to solve this problem, however a model is only good as its data. This is where the real issue is. In the case where more than one disease needs to be diagnosed, a dataset with about 3000 data points (patients), which already took some time to be collected, will not be enough – especially if some portion of the dataset contains missing values. Since different geographic locations have different specificities, as discussed previously, it is hard to merge datasets together to solve this issue.

Onto the end-user themselves. The testing phase of the platform is mainly done by users with a strong background in statistics, computer science, or even machine learning. Even when trying to rely only on the platform’s functionalities, it is hard not to include personal knowledge and previous experience into play. This is why it is paramount to show the platform to end-users with a weaker background in those domains, to make sure that it is still accessible to people with less intuition on what to do and what results actually mean. This is one of the hardest issues to reconcile within the platform. The previous points call for as many functionalities and detail as possible, while the overall goal of the platform calls for simplicity and ease-of-use. A good way to make a compromise between these issues would be to make more complex tools available only if wanted, as an additional method. In parallel, it would be wise to help end-users with less knowledge gain a solid foundation in machine learning by offering some structured information. As previously mentioned, this would be done in the form of a tutorial, use case, and general information on machine learning. In complement to this, a platform manager should be present and address any concerns or issues brought forward by users, as well as be able to explain how to use the functionalities in any possible situation.

RESULTS

After having outlined the tool functionalities, one can now start implementing these. To get the most explanation value out of results, the idea is to take the end user from overall analysis and allow them to thoroughly investigate individual points, as well as global tendencies.

To make sure the functionalities envisioned are both implementable and applicable to medical datasets, a medical dataset was selected and the tools were used on it.

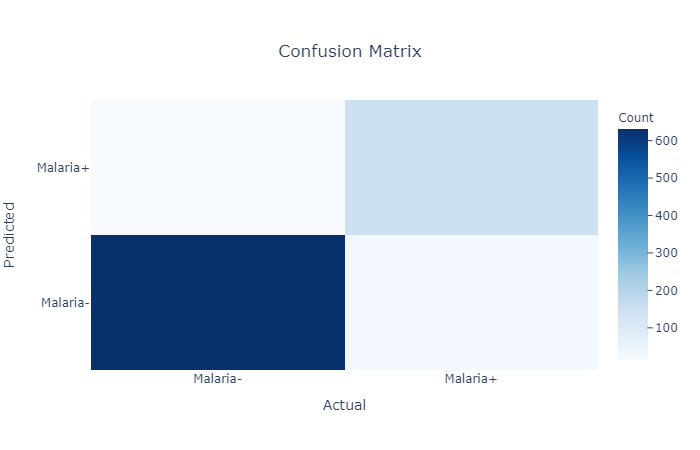
After having trained the model, basic metrics are shown and a confusion matrix is constructed using the sklearn library [TABLE][accuracy/precision/recall][IMAGE}[conf\_mat\_1].

Random forest: simple

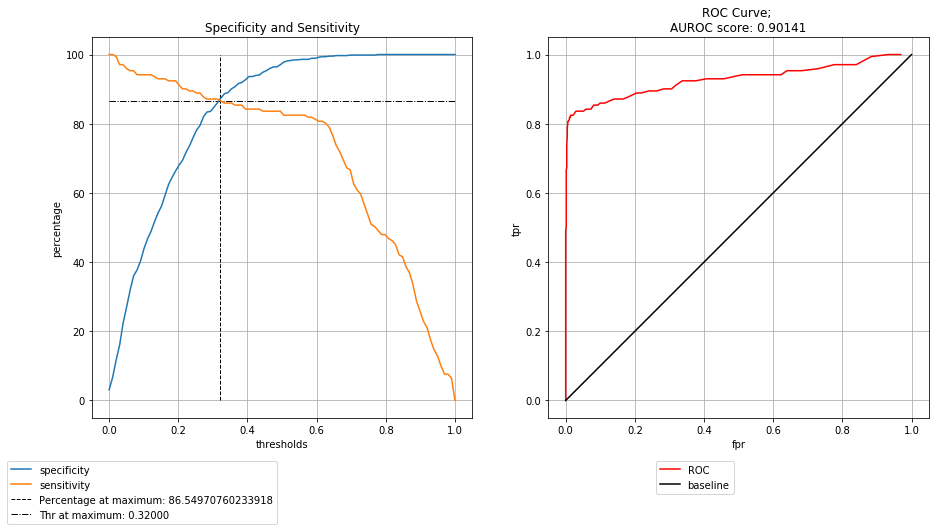
accuracy 0.9460122699386503

precision 0.8245614035087719

recall 0.9096774193548387

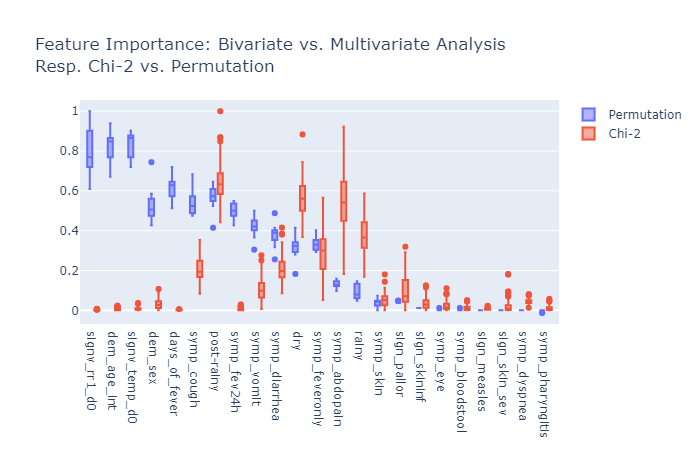


It is necessary to verify that the results from the model prediction are not off-putting before doing some more investigation. While the investigation methods are used to detect any anomalies in the model or data, if the model performs at a poor level, it will be difficult to draw any conclusions. The results show that the model seems to perform well.

A ROC curve and a specificity and sensitivity curve are plotted. [IMAGE][ROC;SENSSPEC]

These plots are commonly used in the medical field.

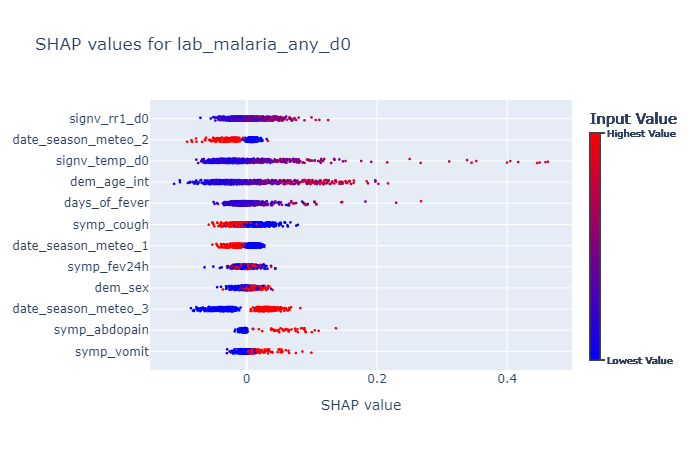
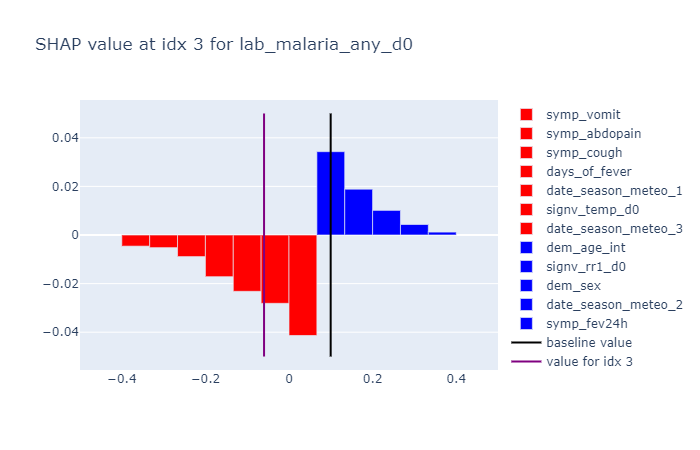
To investigate feature relation to outcome, although the platform previously used partial dependence plots, these will be skipped because of their flaws. Instead, it is recommended to use ALE plots [NOT YET DONE][PROBABLY WONT DO].

Now regarding feature importance, both a bivariate and multivariate test were conducted multiple times through bootstrapping.  [IMAGE][CHI2FEATIMP]

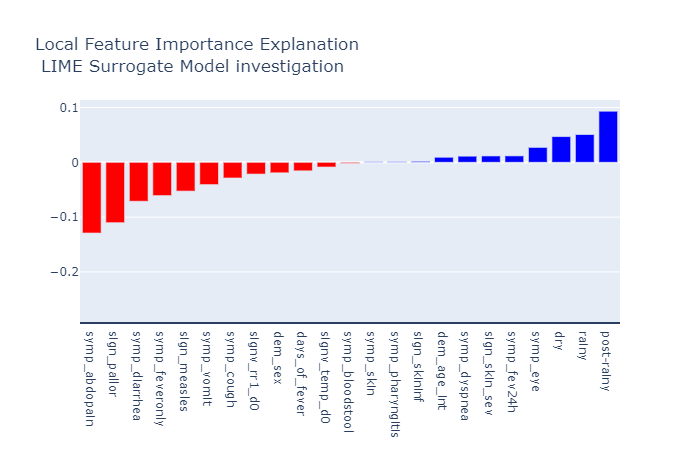
Both have been included for a specific reason. While Chi-2 bivariate analysis is flawed because it does not take into account the other features and possible feature interactions, it is both a common practice in medicine and is therefore widely spread and understood, and a comparison point to the feature importance method. This highlights the importance of using the feature importance metric, or any multivariate analysis rather than a bivariate analysis.

Now that the features have been investigated, it is time to look at some specific points. In order to do this, two methods are used; SHAP and LIME. These figures are made using the python library as a template.

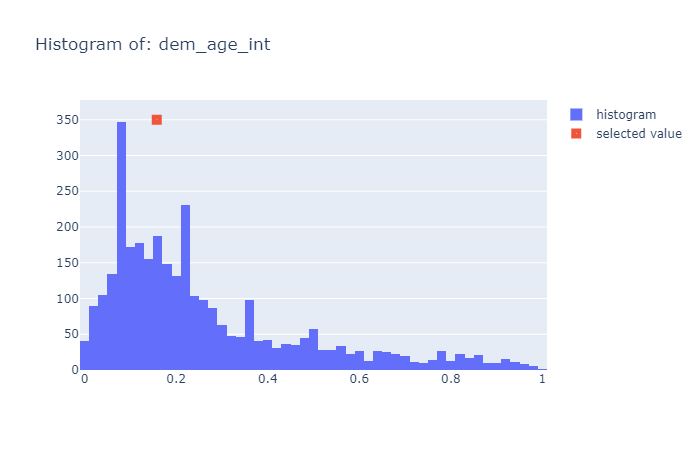
A summary plot is plotted, with the possibility to retrieve the value that needs further investigating from the plot.

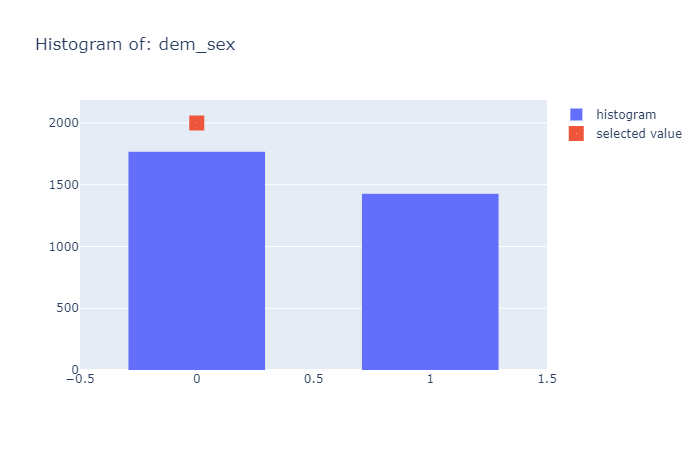
This allows the user to select one point and bring out all the SHAP values concerning this input value. Of course, this is paramount in drawing conclusions.

The baseline value is the mean of all predictions, while the purple line – value for selected index, shows the prediction for that specific input. By taking a look at the color bars, one can understand how the prediction deviates from the mean and is being pushed towards one way or the other. This in effect is a good way to highlight each feature’s independent contribution towards the final outcome.

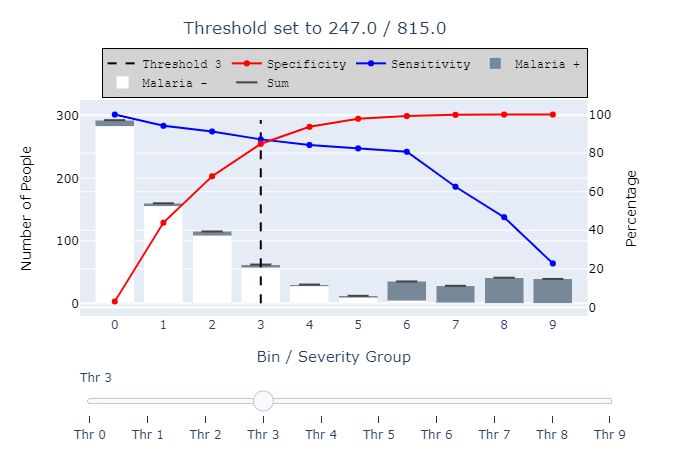
The LIME method allows to perform a similar investigation. [SHOULD PICK SAME IDX SO METHODS CAN BE COMPARED]. This surrogate model interpretation is only valid in neighborhood of the data point selected, while the SHAP plot for one value is only valid for the specific data point. While it may be expected that proximal data points behave similarly, this is a strong assumption and should not be made in this case.

It can then be interesting to see how the input values of that investigated point compare to the rest of the dataset. To this end, the user can take a look at the data distribution of the entire dataset, and place the input value of interest.



This can help the user identify and visualize trends within the outcomes.

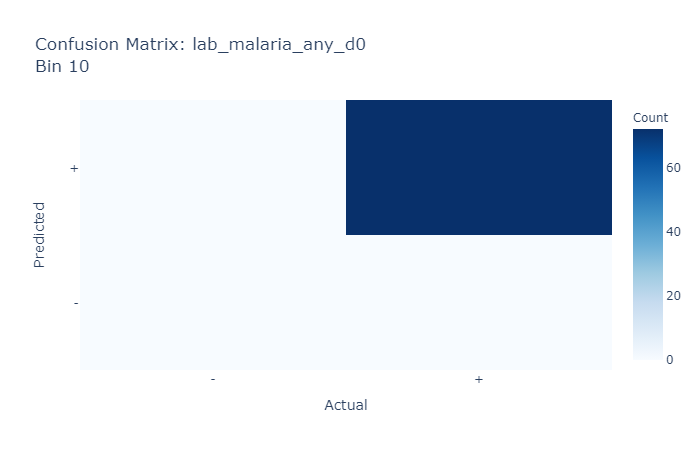
Now that the tools that can be applied to any data and any model have been implemented, a more use-specific tool is constructed. It is templated on a paper by [INCLUDE AUTHORS] which detail the construction of a severity score for malaria patients.

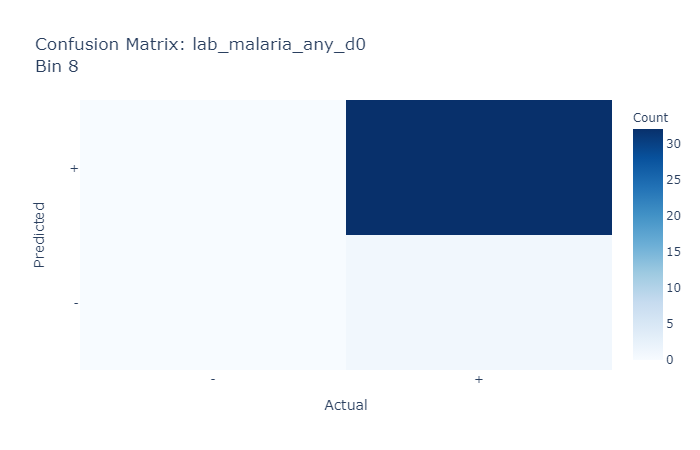
The severity score is built here using the model predictions and their associated probability. The only prediction we are considering is the malaria positive score – therefore the prediction being equal to 1. To each prediction we attach its probability, then bin predictions according to their probability. In the figure above, there are ten bins, which each are separated by thresholds at equal distances; from a probability of 0 to 10%, to 10% to 20%, and onwards. A high associated probability reflects a high possibility of being malaria positive, whereas a low associated probability reflects a low possibility of being malaria positive – in effect a high probability of being malaria negative. These amount respectively to a high severity score and a low severity score. Once these bins have been created, a sensitivity and specificity curve is drawn, with the thresholds used being the ones used for the bin grouping. Once the figure is completed, the user can choose to move a threshold which helps select a total number of patients / data points to include. The threshold represents the total number of people that are at least as high-risk as the bin it is located at. Therefore if there are only a certain of treatment doses available, the user can choose to select the 250 people or so most at risk according to the model.

The bins themselves contain an indication of the labels / ground truth within each bin. The color coding indicates the true values of the predictions. It is thus possible to also quickly identify if it is possible to trust the model, if the bin is split between two colors, it shows the model is indecisive at that threshold.

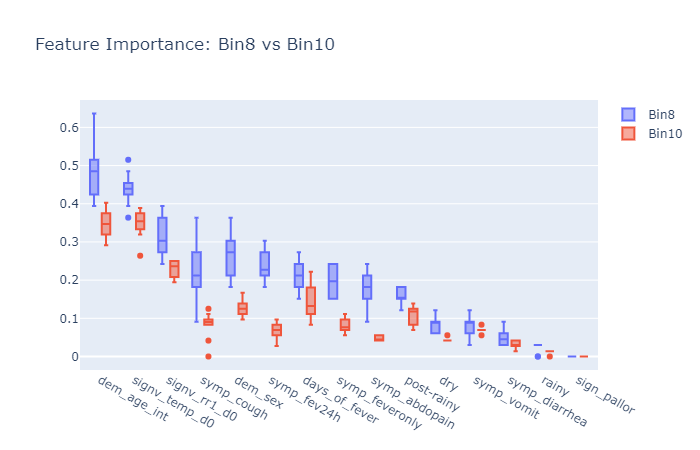
With these bins computed, the analysis above can be refined. By selecting a bin number, the user can then investigate if the model behaved specifically in the way it classified high-severity patients versus low-severity ones, or even between juxtaposed bins of patients.

It can be important to take a look at the confusion matrix of each bin to identify potential weaknesses. In this case, the model seems to perform well according to these metrics.





A tool can be made to compare multiple bins on the same graph. In conjunction with the confusion matrices, an analysis can be made on which features might cause a misclassification.



Effectively, the binning also allows the user to find out which patients are located in which bin, of course.

These tools allow the user to perform this investigation on their own. They can be used on any type of model as the methods used are model-agnostic.