

CRISP-DM Phase Six: Plan Deployment

Lisanne Wallaard
Thesis, LIACS, Leiden University

May 30, 2023

0.1 Introduction

In order to deploy a medical machine learning model as a web app on your local machine, you can use this [framework](#). You just need to download the file from the [repository](#) on GitHub and start adjusting it to your model. In this repository, there is also a [ReadMe](#) explaining the project, giving some information about the used packages and providing the installation steps. In this framework, there are lots of comments explaining the code, but this document will walk you through the steps as well.

0.2 Code Walkthrough

A model can be imported only as a pickle file (Python) or RDS file (R). Enter the path to this model on your device in `PATH_MODEL`.

If your model is developed in R and therefore exported as an RDS file, you should fill in your R path in `PATH_R` as well. You can get your path to R in RStudio by entering `R.home()`. In the main function, you should use the line of code with the `model_rds` variable for importing the model and the other lines of code concerning a pickle file can be deleted.

If your model is developed and exported as a pickle file, you can delete the lines of code related to `PATH_R`. Also the library ‘rpy2’ is not needed as it is only necessary for the use of R models, the same applies to the `r` variable. In the main function, you should import your model using the code line with the `model_pkl` variable and delete the line of code related to an RDS file.

The text and naming of the application can be changed at the beginning of the main function of the file. The comments here are pretty straightforward, so no further explanation is needed. In order to add an image as a page icon, just download the one in the repository and put it in the same folder as `framework.py`. It is also possible to download another image and change the path of `page_icon`.

The input data and input possibilities are loaded using the `input_user()` function. In this function, a few examples are given showing how to get different sorts of input from the user and the corresponding input possibilities in two data frames. The two data frames can be merged together, but if you are not planning

to use the input possibilities during preprocessing, it is also possible to continue with the input data frame only.

Preprocessing of the data is done by the `preprocess_input` function. In this function a few examples of labelling, one hot encoding and scaling are given. However, it depends very much on the preparation of the training data of the model, since the input data should be preprocessed in exactly the same way. After the preprocessing, you might want to reorder the data frame to your preference. It is advised to check the values in the resulting data frame by `st.dataframe(df)`.

There are two buttons added to the application: `submission` and `stop`. When pressing the `stop` button, the application will stop running. After pressing the `submission` button the application will make a prediction. Again you will need to choose the right line of codes using either `model_pkl` or `model_rds`. After `prediction` and `prediction_prob` are calculated, the function call corresponding to your type of file should be selected to print the values.

Finally, there is the option to explain the model with a bar plot showing feature importances or SHAP values. In order to show the feature importances of a model, the `plot` variable should be set equal to `"feature_importance"`. Nevertheless, please note that this is not possible for all R or Python models and there will appear an error if not. Also, different code lines should be selected for the pickle file and RDS file. SHAP values can only be plotted if they are calculated beforehand and exported as a pickle file. The path to this pickle file should be entered in `PATH_SHAP`. If it is not possible or desirable to show a plot, just enter any other word in `plot`.

It is already mentioned a couple of times, but when libraries or lines of code are not relevant to your implementation, they can be removed.

0.3 Conclusion

Hopefully, everything is clear with this code walkthrough and the comments in the framework file. If you, somehow, still experience some difficulties implementing your model as a web app on your machine. It might be handy to take a look at the [experiments](#) folder in the repository. There, multiple implementations of machine learning models in both Python and R, using the same approach as this framework, can be found.