CRISP-DM Phase Six: Plan Deployment

Lisanne Wallaard
Thesis, LIACS, Leiden University

June 19, 2023

1 Introduction

To deploy a medical machine learning model with a straightforward class and probability prediction function 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, many comments explain the code, but this document will also walk you through the steps.

2 Code Walktrough

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 also fill in your R path in `PATH_R`. 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 file's main function. The comments here are pretty straightforward, so no further explanation is needed. 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'. Downloading another image and changing the path of `page_icon` is also possible.

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, 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.

Data preprocessing 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 model's training data since the input data should be preprocessed in precisely 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)`.

Two buttons are added to the application: `submission` and `stop`. The application will stop running when pressing the `stop` button. After pressing the `submission` button, the application will make a prediction. Again, you must 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 file type 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. 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 an error will appear if not. Also, different code lines should be selected for the pickle file and RDS file. SHAP values can only be plotted if 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 string in `plot`.

It is already mentioned a couple of times, but when libraries or lines of code are irrelevant to your implementation, they can be removed.

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 can be found using the same approach as this framework.