WELES, the Model Lake

language and package agnostic base for machine learning models

Wojciech Kretowicz[1], Przemyslaw Biecek[1]

[1]Faculty of Matchematics and Information Science, Warsaw University of Technology

Introduction

Nowadays there exist many tools for creating and training **machine learning models**, each with its own syntax and versions. Thus sharing and reusing once trained models between technologies and over time is hard. The **Model Lake** is an idea of the client server base for **model virtualisation**, where one can upload their model with the information about its **environment**. Then other can use this model regardless of its technology. The **weles** base is our implementation of this idea.

Problem with sharing and production

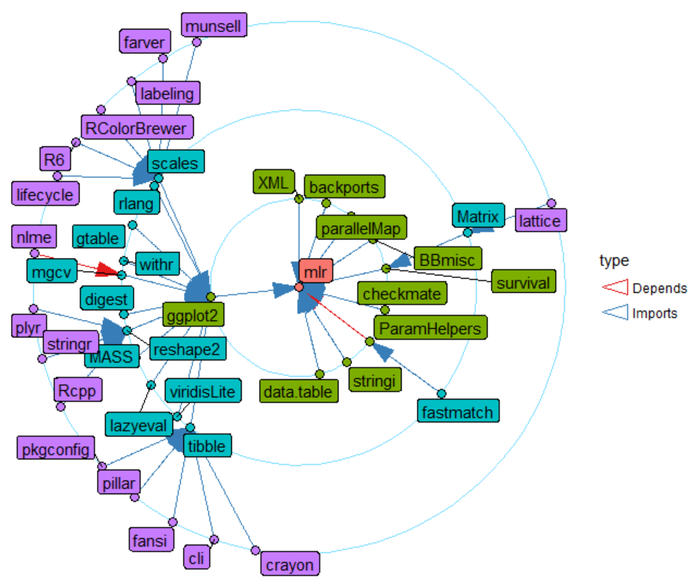

Figure 1: Packages loaded by *mlr*, these need to be restored in session to use model properly (along with needed package itself, i.e. 'ranger')

Both **Python** and **R** are interpreted - such model has got no meaning without a proper interpreter. Furthermore, even the simplest **R** model require bunch of loaded packages. All of them have to be installed to run the model along with proper interpreter with proper version. Thus reusing someone's model on your computer is very inconvenient because you need to restore whole its environment.

Workflow

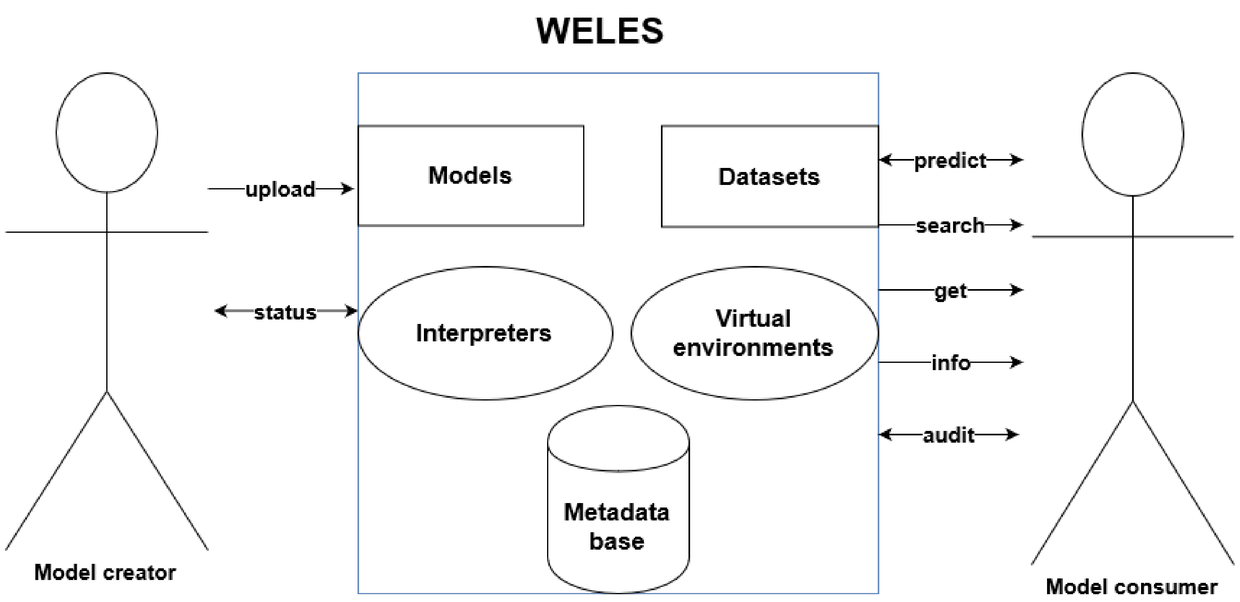

Figure 2: Architecture of the weles system. The **model repository** has five components, repository for binary versions of models, binary versions of datasets, SQL database for metadata and collection of interpreters (**R/Python**) along with specific **virtual environments**

Basically there are two actors. First is called **Model creator** who using the language **A** and packages **P** creates and trains model. Then creator decides to share it in the **weles**. He uses **Python** or **R** client package to upload the binary version of the model along with information about its **environment** and dataset used to training.
Then proper virtual environment in **weles** is created (or if already created, reused). After uploading another user, using language **B** and packages **Q** can freely reuse this model.

Code examples in R and Python

```
model_upload(mod, 'model_name',
'This is an example model', 'target', c('tag1', 'tag2'), data,
'example_data', 'This is an example data')
model.predict('model_name', data)
```

Sequence Diagrams

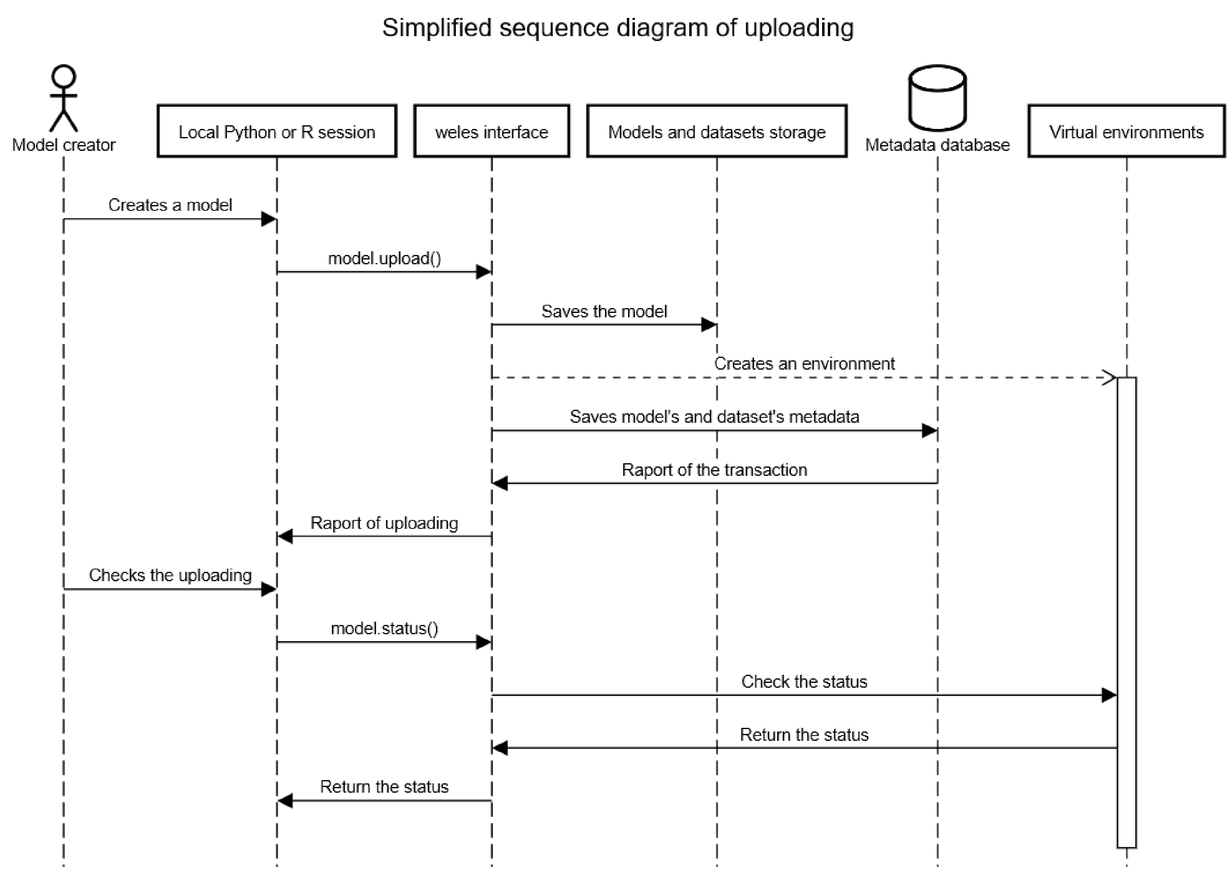

Figure 3: An example sequence diagram for model uploading. **Model creator** trains a model in its client **R/Python** session and then may upload the model to the **weles** database. The binary version of the model is stored and also a **virtual environment** is prepared with all dependencies present in the client session

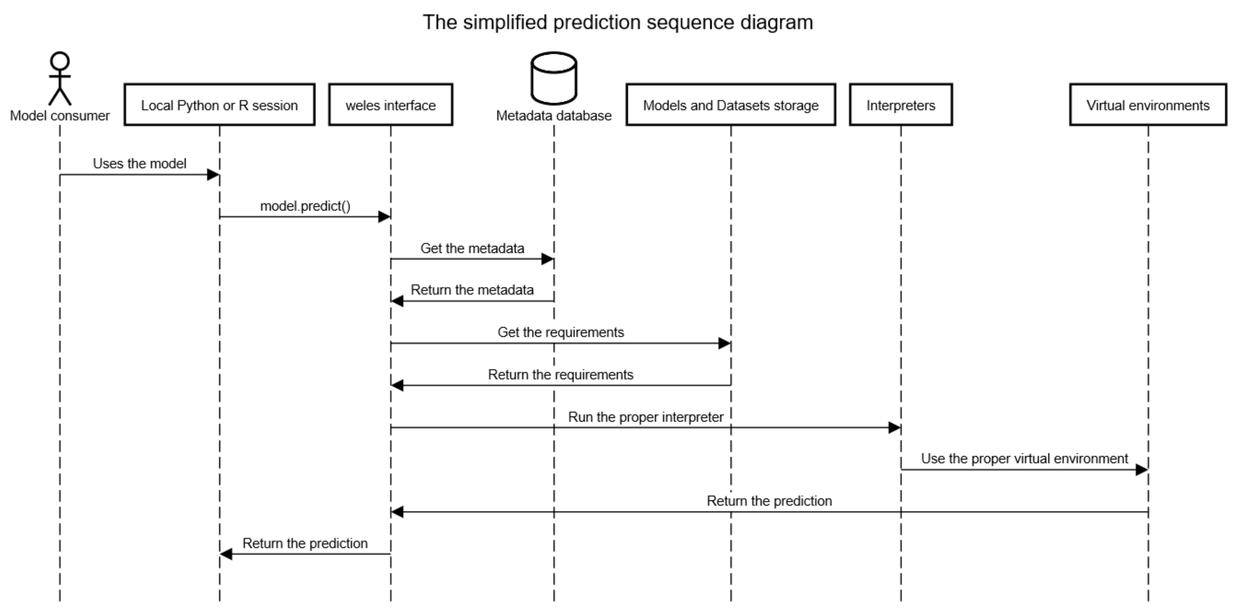

Figure 4: An example sequence diagram for model predictions. Model is executed on the server side. All required dependencies were prepared during the model uploading phase, see Figure 3. Note that server side virtual environment may be different than client side environment. The **model consumer** can call an **R model from Python or Python model from R**.

Docker

Configuring this base may be confusing. To allow you easy installation and hosting such **Model Lake** by yourself, we prepared proper **Dockerfile**, which you can download from our Github. Then you can easily build your own image of **weles** specifying for example your master password or interpreters you would like to have.

References

[1] Machine learning requires a fundamentally different deployment approach. https://www.oreilly.com/radar/machine-learning-requires-a-fundamentally-different-deployment-approach/, 2019.

[2] Przemysław Biecek and Marcin Kosiński. archivist: An r package for managing, recording and restoring data analysis results. *Journal of Statistical Software, Articles*, 82(11):1–28, 2017. ISSN 1548-7660. doi: 10.18637/jss.v082.i11. URL https://www.jstatsoft.org/v082/i11.

Acknowledgements

This work was financially supported by Polish Centre for Research and Development (Grant POIR.01.01.01-00-0328/17).

- ✉ wojtekkretowicz@gmail.com
- ✉ https://github.com/ModelOriented/weles