

Patient-Level Prediction Installation Guide

Jenna Reys, Peter R. Rijnbeek

2020-06-03

Contents

1	Introduction	1
2	Software Prerequisites	1
2.1	Windows Users	1
2.2	Mac/Linux Users	1
3	Installing the Package	2
3.1	Installing PatientLevelPrediction using drat	2
3.2	Installing PatientLevelPrediction using devtools	2
4	Creating Python Reticulate Environment	2
5	Testing installation	2
6	Installation issues	3
7	Acknowledgments	3

1 Introduction

This vignette describes how you need to install the Observational Health Data Sciences and Informatics (OHDSI) `PatientLevelPrediction` package under Windows, Mac, and Linux.

2 Software Prerequisites

2.1 Windows Users

Under Windows the OHDSI Patient Level Prediction (PLP) package requires installing:

- R (<https://cran.cnr.berkeley.edu/>) - (R \geq 3.3.0, but latest is recommended)
- Rstudio (<https://www.rstudio.com/>)
- Java (<http://www.java.com>)
- RTools (<https://cran.r-project.org/bin/windows/Rtools/>)

2.2 Mac/Linux Users

Under Mac and Linux the OHDSI Patient Level Prediction (PLP) package requires installing:

- R (<https://cran.cnr.berkeley.edu/>) - (R \geq 3.3.0, but latest is recommended)
- Rstudio (<https://www.rstudio.com/>)
- Java (<http://www.java.com>)

- Xcode command line tools(run in terminal: `xcode-select --install`) [MAC USERS ONLY]

3 Installing the Package

The preferred way to install the package is by using `drat`, which will automatically install the latest release and all the latest dependencies. If the `drat` code fails or you do not want the official release you could use `devtools` to install the bleeding edge version of the package (latest master). Note that the latest master could contain bugs, please report them to us if you experience problems.

3.1 Installing PatientLevelPrediction using drat

To install using `drat` run:

```
install.packages("drat")
drat::addRepo("OHDSI")
install.packages("PatientLevelPrediction")
```

3.2 Installing PatientLevelPrediction using devtools

To install using `devtools` run:

```
install.packages('devtools')
devtools::install_github("OHDSI/FeatureExtraction")
devtools::install_github('ohdsi/PatientLevelPrediction')
```

When installing using `devtools` make sure to close any other Rstudio sessions that are using `PatientLevelPrediction` or any dependency. Keeping Rstudio sessions open can cause locks that prevent the package installing.

4 Creating Python Reticulate Environment

Many of the classifiers in the `PatientLevelPrediction` use a Python back end. To set up a python environment run:

```
library(PatientLevelPrediction)
reticulate::install_miniconda()
configurePython(envname='r-reticulate', envtype='conda')
```

To add the R keras interface, in Rstudio run:

```
devtools::install_github("rstudio/keras")
library(keras)
install_keras()
```

Some of the less frequently used classifiers are not installed during this set-up to add them run:

For GBM survival:

```
reticulate::conda_install(envname='r-reticulate', packages = c('scikit-survival'), forge = TRUE, pip = F)
```

For any of the torch models:

```
reticulate::conda_install(envname='r-reticulate', packages = c('pytorch', 'torchvision', 'cpuonly'), fo
```

5 Testing installation

To test whether the package is installed correctly run:

```
library(DatabaseConnector)
connectionDetails <- createConnectionDetails(dbms = 'sql_server',
                                             user = 'username',
                                             password = 'hidden',
                                             server = 'your server',
                                             port = 'your port')
PatientLevelPrediction::checkPlpInstallation(connectionDetails = connectionDetails,
                                             python = T)
```

To test the installation (excluding python) run:

```
library(DatabaseConnector)
connectionDetails <- createConnectionDetails(dbms = 'sql_server',
                                             user = 'username',
                                             password = 'hidden',
                                             server = 'your server',
                                             port = 'your port')
PatientLevelPrediction::checkPlpInstallation(connectionDetails = connectionDetails,
                                             python = F)
```

The check can take a while to run since it will build the following models in sequence on simulated data: Logistic Regression, RandomForest, MLP, AdaBoost, Decision Tree, Naive Bayes, KNN, Gradient Boosting. Moreover, it will test the database connection.

6 Installation issues

Installation issues need to be posted in our issue tracker: <http://github.com/OHDSI/PatientLevelPrediction/issues>

The list below provides solutions for some common issues:

1. If you have an error when trying to install a package in R saying ‘**Dependency X not available ...**’ then this can sometimes be fixed by running `install.packages('X')` and then once that completes trying to reinstall the package that had the error.
2. I have found that using the github devtools to install packages can be impacted if you have **multiple R sessions** open as one session with a library open can cause the library to be locked and this can prevent an install of a package that depends on that library.

7 Acknowledgments

Considerable work has been dedicated to provide the PatientLevelPrediction package.

```
citation("PatientLevelPrediction")

##
## To cite PatientLevelPrediction in publications use:
##
## Reips JM, Schuemie MJ, Suchard MA, Ryan PB, Rijnbeek P (2018). "Design and
## implementation of a standardized framework to generate and evaluate patient-level
## prediction models using observational healthcare data." _Journal of the American
## Medical Informatics Association_, *25*(8), 969–975. <URL:
## https://doi.org/10.1093/jamia/ocy032>.
##
## A BibTeX entry for LaTeX users is
##
```

```
## @Article{
##   author = {J. M. Reps and M. J. Schuemie and M. A. Suchard and P. B. Ryan and P. Rijnbeek},
##   title = {Design and implementation of a standardized framework to generate and evaluate patient-
##   journal = {Journal of the American Medical Informatics Association},
##   volume = {25},
##   number = {8},
##   pages = {969-975},
##   year = {2018},
##   url = {https://doi.org/10.1093/jamia/ocy032},
## }
```

Please reference this paper if you use the PLP Package in your work:

Reps JM, Schuemie MJ, Suchard MA, Ryan PB, Rijnbeek PR. Design and implementation of a standardized framework to generate and evaluate patient-level prediction models using observational healthcare data. J Am Med Inform Assoc. 2018;25(8):969-975.

This work is supported in part through the National Science Foundation grant IIS 1251151.