

Patient-Level Prediction Installation Guide

Jenna Reys, Peter R. Rijnbeek

2018-10-04

Contents

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

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 >= 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/>)
- Anaconda 3.6 (<https://www.anaconda.com/download>) - this will require checking your path variable to ensure the correct python is used by R - more instructions below. For Python you need to make sure it is in the Path: go to my computer -> system properties -> advanced system settings Then at the bottom right you'll see a button: Environmental Variables, clicking on that will enable you to edit the PATH variable to add the Anaconda location. In R you need to check the Path is correct: You can access the path variable in R using `Sys.getenv('PATH')`. This should contain the location of your Anaconda or python 3.6.
- If you have Anaconda and want to use PyTorch v0.4 (<https://pytorch.org>) as the backend of deep learning, you can directly use command “conda install pytorch torchvision -c pytorch” for Linux. Please refers to commands for installing PyTorch (<https://pytorch.org>) on other develop environments.
- To add the R keras interface, in Rstudio run:

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

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 >= 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]
- Python 3.6 (<https://www.python.org/downloads/>) - this will require checking your path variable to ensure this version python is added - more instructions below
- To add the R keras interface, in Rstudio run:

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

2.2.1 Setting up Python for Mac/Linux Users

After installing python 3.6 check it is working by typing `python3` to open python in a terminal.

To get the package dependencies, in a terminal run:

```
pip3 install --upgrade pip
pip3 install -U NumPy
pip3 install -U SciPy
pip3 install -U scikit-learn
pip3 install -U torch
pip3 install --upgrade tensorflow
pip3 install keras
```

Dependent on your permissions you may need to add a `sudo` command in front of the `pip3` commands.

Mac and Linux users need edit the bash profile to add python in their Path by running in the terminal: `touch ~/.bash_profile; open ~/.bash_profile;` and adding in the location of python 3.6 in the PATH variable. You can find the location of the python versions by typing this in a terminal:

```
type -a python
```

Furthermore, you need to specify in their R environment that R needs to use python 3.6 rather than the default python. In a new Rstudio session run this to open the environment file:

```
install.packages('usethis')
usethis::edit_r_environ()
```

In the file that opens add and save: `PATH= {The path containing the python 3}`

```
USESPECIALPYTHONVERSION="python3.6"
```

You now need to compile PythonInR so it uses python 3.6. In a new R studio session run:

```
Sys.setenv('USESPECIALPYTHONVERSION='python3.6')
devtools::install_bitbucket("Floooo/PythonInR")
```

This should now set the PythonInR package to use your python 3.6. Please note: if you update the path while R is open, you will need to shutdown R and reopen before the path is refreshed.

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")

## Installing PatientLevelPrediction using devtools
To install using devtools run:
```

```
install.packages("devtools")
library("devtools")
install_github("ohdsi/SqlRender")
install_github("ohdsi/DatabaseConnectorJars")
install_github("ohdsi/DatabaseConnector")
install_github("ohdsi/FeatureExtraction")
install_github("ohdsi/OhdsiSharing")
install_github("ohdsi/OhdsiRTools")
install_github("ohdsi/BigKnn")
install_github("ohdsi/PatientLevelPrediction")
```

4 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.

5 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.

6 Acknowledgments

Considerable work has been dedicated to provide the PatientLevelPrediction package.

```
citation("PatientLevelPrediction")
```

```
##
## Jenna Reps, Martijn J. Schuemie, Marc A. Suchard, Patrick B.
## Ryan and Peter R. Rijnbeek (2018). PatientLevelPrediction:
## Package for patient level prediction using data in the OMOP
## Common Data Model. R package version 2.0.5.
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
##   title = {PatientLevelPrediction: Package for patient level prediction using data in the OMOP Com
## Model},
##   author = {Jenna Reps and Martijn J. Schuemie and Marc A. Suchard and Patrick B. Ryan and Peter R.
##   year = {2018},
##   note = {R package version 2.0.5},
## }
```

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.