Using the package skeleton for validating existing model studies

Jenna M. Reps

2020-10-14

Contents

1 Introduction

This vignette describes how one can use the package skeleton for validating existing prediction model studies on any data in the OMOP CDM. The package can be used to execute the study at any site that has access to an observational database in the Common Data Model. It will perform the following steps:

1. Instantiate all cohorts needed for the study in a study-specific cohort table.
2. The main analysis will be executed using the `PatientLevelPrediction` package, which involves applying and validating existing prediction models.
3. The results can be modified to remove sensitive data ready for sharing.

1.1 Option 1 (recommended): Installing package via GitHub

The easiest way to install a study package is to run the code but replace '' with the name of the study you are downloading from GitHub:

```
install.packages('devtools')
devtools::install_github('ohdsi-studies/<studyName>')
```

This will then start to install the package. It may require installing dependancies. If the install is successfull it will end with 'done()'. If you have a error during install it will not install and you need to determine the error, fix it and then try again. If you run into issues please contact us using the OHDSI forum or in the GitHub issues tab.

1.2 Optim 2: Installing package by building it locally

To install a package locally you can populate the skeleton package or download a GitHub study package and then build it by:

1.2.1 Open the project in Rstudio

Make sure to have RStudio installed. Then open the R project downloaded from ATLAS by decompressing the downloaded folder and clicking on the .Rproj file (where is replaced by the study name you specified). This should open an RStudio session.

1.2.2 Installing all package dependencies

Before you can build the package you downloaded you need to make sure you have all the dependencies:

```
source('./extras/packageDeps.R')
```

1.2.3 Building the package

Once you have the dependencies installed you can now build the R package. This creates a library you can load to run the validation study. To build the package click 'Build' on the top right hand side tab menu (there are tabs: 'Environment', 'History', 'Connections', 'Build', 'Git'). Once in 'Build' click the 'Install and Restart' button. This will now build your package and create the R library. If it succeeds you will see '* DONE ()', if it fails you will see red output and the library may not be created. Please report an issue to: https://github.com/OHDSI/PatientLevelPrediction/issues if your library does not get created.

1.3 Running the package

To run the study, open the extras/CodeToRun.R R script (the file called `CodeToRun.R` in the `extras` folder). This folder specifies the R variables you need to define (e.g., outputFolder and database connection settings). See the R help system for details:

```
library(<studyName>)
?execute
```

By default all the options are set to F for the execute fuction:

```
# The folder where the study intermediate and result files will be written:
outputFolder <- "./<studyName>"

# Details for connecting to the server:
dbms <- "you dbms"
user <- 'your username'
pw <- 'your password'
server <- 'your server'
port <- 'your port'

connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = dbms,
                                                                server = server,
                                                                user = user,
                                                                password = pw,
                                                                port = port)

# Add the database containing the OMOP CDM data
cdmDatabaseSchema <- 'cdm database schema'
# Add a database with read/write access as this is where the cohorts will be generated
cohortDatabaseSchema <- 'work database schema'

oracleTempSchema <- NULL

# table name where the cohorts will be generated
cohortTable <- '<studyName>Cohort'

# TAR settings
sampleSize <- NULL
riskWindowStart <- 1
startAnchor <- 'cohort start'
riskWindowEnd <- 365
```

```
endAnchor <- 'cohort start'
firstExposureOnly <- F
removeSubjectsWithPriorOutcome <- F
priorOutcomeLookback <- 99999
requireTimeAtRisk <- F
minTimeAtRisk <- 1
includeAllOutcomes <- T


#======================

execute(connectionDetails = connectionDetails,
                                    cdmDatabaseSchema = cdmDatabaseSchema,
                                    cdmDatabaseName = cdmDatabaseName,
                                    cohortDatabaseSchema = cohortDatabaseSchema,
                                    cohortTable = cohortTable,
                                    sampleSize = sampleSize,
                                    riskWindowStart = riskWindowStart,
                                    startAnchor = startAnchor,
                                    riskWindowEnd = riskWindowEnd,
                                    endAnchor = endAnchor,
                                    firstExposureOnly = firstExposureOnly,
                                    removeSubjectsWithPriorOutcome = removeSubjectsWithPriorOutcome,
                                    priorOutcomeLookback = priorOutcomeLookback,
                                    requireTimeAtRisk = requireTimeAtRisk,
                                    minTimeAtRisk = minTimeAtRisk,
                                    includeAllOutcomes = includeAllOutcomes,
                                    outputFolder = outputFolder,
                                    createCohorts = T,
                                    runAnalyses = T,
                                    viewShiny = T,
                                    packageResults = F,
                                    minCellCount= 5,
                                    verbosity = "INFO",
                                    cdmVersion = 5)
```

If you run the above nothing will happen as each option is false. See the table below for information about each of the inputs.

Input	Description	Example
connectionDetails	The details to connected to your OMOP CDM database - use DatabaseConnector package's createConnectionDetails()	createConnectionDetails(dbms = 'postgresql', server = 'database server', user = 'my username', password = 'donotshare', port = 'database port')
cdmDatabaseSchema	The schema containing your OMOP CDM data	'my_cdm_data.dbo'
cdmDatabaseName	A shareable name for the OMOP CDM data	'My data'
oracleTempSchema	The temp schema if dbms = 'oracle' - NULL for other dbms	'my_temp.dbo'
cohortDatabaseSchema	The schema where you have an existing cohort table or where the package will create a cohort table and insert the study cohorts	'scratch.dbo'

Input	Description	Example
cohortTable	The table name where you cohorts will be written (if creating the cohort pick an unused table name)	'myTable'
sampleSize	Sample from the target population]	NULL
riskWindowStart	The time at risk starts this many days after the endAnchor	1
startAnchor	Make the time-at-risk start relative to cohort start or cohort end	'cohort start'
endAnchor	Make the time-at-risk end relative to cohort start or cohort end	'cohort start'
firstExposureOnly	If a patient in the target population at different times restrict to first time?	T
removeSubjectsWithPriorOutcome	Remove people with the outcome before index?	T
priorOutcomeLookback	Time to look back from index if removeSubjectsWithPriorOutcome = T	9999
minTimeAtRisk	Minimum time at risk a patient must satisfy to be in the target population (used when requireTimeAtRisk = T)	1
includeAllOutcomes	Whether to keep people with the outcome during TAR even if they dont have complete follow-up	F
outputFolder	The location where the results of the study will be saved	'C:/amazingResults'
createCohorts	TRUE or FALSE indicating whether to create the target population and outcome cohorts for the study	TRUE
runAnalyses	TRUE or FALSE indicating whether to run the study analysis - developing and internally validating the models	TRUE
packageResults	TRUE or FALSE indicating whether to remove sensitive counts (determined by the minCellCount input) or sensitive information from the results and creates a zipped file with results that are safe to share (saved to the outputFolder location). Note: This requires running the study successfully first.	TRUE
minCellCount	integer that determines the minimum result count required when sharing the results. Any result table cells with counts < minCellCount are replaced with -1 to prevent identification issues with rare diseases	10
viewShiny	TRUE or FALSE indicating whether to view a shiny app with the results from the study, Note: This requires running the study successfully first.	TRUE

To create the target and outcome cohorts (cohorts are created into cohortDatabaseSchema.cohortTable)

```
createCohorts = T
```

To validate the models in the study run the code:

```
runAnalyses = T
```

If the study runs and you get results, you can then interactively explore the results by running:

```
viewShiny= T
```

To package the results ready for sharing with others you can set:

```
packageResults = T
```

1.4 Results

After running the study you will find the result in the `[outputFolder]/cdmDatabaseName` directory as an rds object named 'validationResult.rds'.

The validationResult object is a list containing:

Object	Description	Edited by packageResult
`inputSetting`	All the settings required to reproduce the study	Yes - passwords and database settings are removed
`executionSummary`	Information about the R version, PatientLevelPrediction version and execution platform info	No
`model`	The trained model	No
`analysisRef`	Used to store a unique reference for the study	No
`covariateSummary`	A dataframe with summary information about how often the covariates occured for those with and without the outcome	Yes - minCellCounts censored
`performanceEvaluation$ evaluationStatistics`	Performance metrics and sizes	No
`performanceEvaluation$ thresholdSummary`	Operating characteristcs @ 100 thresholds	Yes
`performanceEvaluation$ demographicSummary`	Calibration per age group	Yes
`performanceEvaluation$ calibrationSummary`	Calibration at risk score deciles	Yes
`performanceEvaluation$ predictionDistribution`	Distribution of risk score for those with and without the outcome	Yes

1.5 extras/PackageMaintenance.R

This file contains other useful code to be used only by the package developer (you), such as code to generate the package manual, and code to insert cohort definitions into the package. All statements in this file assume the current working directory is set to the root of the package.