Using the package skeleton for external validation studies

Jenna M. Reps

2020-03-09

Contents

1 Introduction

This vignette describes how one can use the package skeleton for validating patient-level prediction studies to create one's own study package. This skeleton is aimed at external validation studies using the `PatientLevelPrediction` package. The resulting package can be used to execute the external validation 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 evaluating the performance of one or many previously developed models.
3. The results can be packaged up (after removing sensitive data) ready to share with the study co-ordinator

1.1 Open the project in Rstudio

Skip this step if you are just running the study via github (skip to ' Running the package')

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 in ATLAS). This should open an RStudio session.

1.2 Building the package

Skip this step if you are just running the study via github (skip to ' Running the package')

First you need to 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

If running the study from github you first need to install the package:

```
# To install the package from github:
install.packages("devtools")
devtools::install_github("OHDSI-studies/SkeletonValidationStudy")
```

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(SkeletonValidationStudy)
?execute
```

The inputs to the execute function for validating prediction models are described below:

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'
databaseName	A shareable name for the OMOP CDM data being used to validate the models	'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'
cohortTable	The table name where you cohorts will be written (if creating the cohort pick an unused table name)	'myTable'
outputFolder	The location where the results of the study will be saved - if you also developed the model you can set this to the Validation folder where your model development results were saved	'C:/predictingMI/Validation'
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

Input	Description	Example
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
sampleSize	An integer > 0 specfiying the size of a sample of patients to extract from the target cohort. The model will only be validated on the sample - this is useful when the target cohort is large and you have limited time	1000000
keepPrediction	TRUE or FALSE indicating whether to save the individual predictions when applying the models to the target cohort (or sample)	TRUE

To create the target and outcome cohorts (cohorts are created into cohortDatabaseSchema.cohortTable) make sure createCohorts is set to TRUE

```
createCohorts = T
```

To externally validate the models make sure runAnalyses is set to TRUE:

```
runAnalyses = T
```

To package the results ready for sharing with others you can set packageResults to TRUE. This will only run if you have previously ran the analysis and have results:

```
packageResults = T
```

1.4 Results

After running the study you will find the results in the specified `outputFolder` directory. The `outputFolder` directory will contain a folder for each database you used to externally validate the models. For example, suppose you ran the study on two databases that you set databaseName as 'bestData' and 'secondBestData', then you would have two folders in `outputFolder`:

- bestData
- secondBestData

Then these folders would contain folders for each model validated. Lets assume you valdiated 3 models, then you would have the follow saved in `outputFolder`:

- bestData
  - Analysis_1
  - Analysis_2
  - Analysis_3
- secondBestData
  - Analysis_1
  - Analysis_2
  - Analysis_3

Each of the 'Analysis_i' folders contain a validationResult.rds object. This object contains the results of externally validating model i. For example, you can load the result of the model 2 when applied to 'bestData' with:

```
validationResult <- readRDS(file.path(outputFolder, 'bestData', 'Analysis_2', 'validationResult.rds
```

The validationResult.rds object is a list containing:

Object	Description	Edited by packageResult
`inputSetting`	The inputs such as cohort ids	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
`prediction`	A dataframe with information about the target cohort and the prediction scores - only kept if keepPrediction = TRUE	Yes - removed when sharing
`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

When you package the result the validationResult.rds is modified to remove any sensitive data that should not be shared (see the table indicating which outputs are modified by the packageResults). The input 'minCellCount' is used when packaging the results. The ready to share results are saved as a compressed folder '[outputFolder]/[databaseName].zip'. In addition, for some operating systems (that can not unlink the temporary export folder) you will also find rds files 'validationResult.rds' in 'Analysis_i' folders at the location: '[outputFolder]/[databaseName]/export'.

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.