Assignment 4 DATA423-20S1

Zhen Huang

5/22/2020

Question 1-Caret

Overview

Caret The `caret` package which described, by its first author/maintainer Max Kuhn, as a unified interface of predictive models. To be specific, it provides a set of tools (functions and classes) that attempt to streamline the process for creating classification and regression models.[1] To achieve this, the package contains tools such as data splitting, pre-processing and resampling, feature selection, model tuning, performance evaluation, models comparation, visualization for the models and the other funtionality.

Python's scikit-learn Scikit-learn is an open source machine learning library in Python that supports supervised and unsupervised learning. Similar to `caret`, it also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.[2]

R's mlr3 package One of the design princles of `mlr3` is focusing on computation. Together, the `mlr3` package and its ecosystem provide a generic framework for machine learning tasks for the R language. Similar to `caret` wrapping function `caret::train`, `mlr3` also provide a unified interface to many popluar machine learing algorithms in R, which are objects of class `mlr3::learner`. [3]

Compare and Contrast the Frameworks

Tasks coverage The general data science tasks were taken into account in a comparison of functions coverage, which is shown in the below table[4].

Tasks Coverage	`caret`	`scikit-learn`	`mlr3`
Visualisation	*	`matplot`	`viz`
Pre-processing	* or work with `recipe`	* work with `numpy`	`pipelines`
Data spliting	*	*	`pipelines`
Standardised training/predicting	wrapping	*	wrapping
Feature engineering/selection	*	*	`filter`,`fswrap`
Method categorisation	*		*
Resamplling	*	*	*
Hyperparameter tuning	*	*	`tuning`,`hyperband`,`mbo`,`paradox`
Model evaluation	*	*	`measures`
Model selection	*	*	
Parallel Processing	*	*	*

[1] The caret Package http://topepo.github.io/caret/index.html
[2] scikit-learn.org https://scikit-learn.org/stable/getting_started.html
[3] mlr3 Manual https://mlr3book.mlr-org.com/introduction.html
[4] *(asterisk) means functions or classes intergrated in the package

Except for method categorisation and model selection, all three packages can handle most of the tasks in predicting models. There is much difference in details worthing to discuss, such as the way the packages achieving the tasks and specific functions or classes within these packages.

For the vasualisation task, `caret` offers its own functions which suitable to the specific model, `scikit-learn` leaves this part of job to `matplot` and `mlr3` offers through its extension packge `viz`. This kind of extension functions make up the `mlr3` ecosystem, which is designed by following the principles in the `mlr3`.

It is critical for predicting a good model by implementing an efficient and reasonable pre-processing step dealing with missing data, variable encoding and various feature transformations. Although three packages have their own tools, it is usually a good practice working together with other packages, such as `recipe` and `caret`, `numpy` and `scikit-learn`. There is a similar idea of `pipelines` in `mlr3` and Pipeline APIs in Pyspark machine learning package, furthermore, `pipelines` in `mlr3` offers an more complicated structure by networking the pre-processing steps together in `Graph`, which is not only a visualisation for the inputs and outputs but making it possible to connect the steps of training/predicting/hyperparameter tuning with pre-processing together.

Both `caret` and `mlr3` wrap the methods of other packages in R, while `scikit-learn` implements method in itself. So, depending packages need to be loaded when training/predicting models in `caret` and `mlr3` explicitly or implicitily.

Models A simple comparison with the number of methods in three packages is showed below. [5] [6] [7]

	`caret`	`scikit-learn`	`mlr3`
Number of methods	238	140	51

The very new comer is `mlr3`, lauched from 2019, which supports less models than the other two packges. Under these simple numbers, `caret` focuses on the classification and regression methods, just as its acronym. Note that only `scikit-learn` supports cluster methods at the present, while `mlr3` is going to achieve this through its extension package `mlr3cluster` in progress. So if we are facing an unsupervied clustering problem, `scikit-learn` is the most approperiate choice.

Some features that cannot be listed in the table in details: model number 238 in caret (CR), custom model tuning: random search resampling: up/down/weight/custom measuring performance generally, specific steps in task(categories)

ecosystem: recipes/caret scikit, mlr3 ecosystem platform: python spark/ r publish

Some ideas you might like to explore:

What are the strengths and weaknesses of each framework? cares wrapper, unifier, model number, visualization, parallise Caret is not a framework for every data science problem. It is not suitable for: • Big data • Deep learning • Image segmentation/classification • Video/audio data • Unsupervised learning e.g. Cluster analysis & anomoly detection

What ideas are common across all frameworks?

Are certain problems better solved in one framework versus another?

Question 2 - Report assessment (Marks: 1/3)

In the resources for Assignment 4 is a report on a regression investigation, "Ass4 Report.pdf". There are a number of instances of not-best-practice present in this report. There are also some ethical issues to consider.

[5] http://topepo.github.io/caret/available-models.html
[6] https://scikit-learn.org/stable/user_guide.html
[7] https://mlr3book.mlr-org.com/list-learners.html

Can you identify them. A list of issues is required with a sentence describing each issue. You are allocated marks for each valid point (that is not trivial or a restatement of a previous point). You will lose points for invalid points.

When answering this question, imagine a colleague has written this report and has asked you to vet it before it is delivered to the Human Resources department. Your job is to comment on all the things that can and should be improved.

Example sentences:

"Only 1 method has been attempted; there is no evidence the method is the best possible for the data."

"The Nominal variable 'agCode' has high cardinality and should be encoded using a method that does not introduce too many numeric columns."

"The report does not discuss the imbalance of males to females in the data. Is it ethical to have a single model for both males and females?"

Hint: I managed to find 30 issues.

Question 3 - Quality Control (Marks: 1/3)

The file monitor.csv contains comma separated data. The columns are

Timestamp - the timestamp of a model prediction being run ProcessMemory - the allocated memory (MB) of the relevant server process Prediction - the value predicted by the model PredictionTimeMS - the duration of the prediction task in milliseconds Using the supplied CSV data, generate control charts and answer the following questions:

a) Is the memory usage of the server in control?

b) Is the prediction time of the model in control?

c) Is the stream of predictions in control?

The relevant control charts would be "xbar" and "s". You should aggregate the data per day. Assume the first 20 days of data can be used to establish the control limits for the remainder of the data.

Remember to report upon runs.signal and sigma.signal from each chart summary.

Have a read of https://cran.r-project.org/web/packages/qicharts/vignettes/controlcharts.html Present your charts and results as an R-Notebook document. In RStudio use the menu choices: File / New File / R Notebook.