Tutorial for using conformal prediction in KNIME

Introducing Conformal Regression in KNIME

Tuwe Löfström, Artem Ryasik, Ulf Johansson

23.08.2022

✉ tuwe.lofstrom@ju.se

⌂ https://ju.se/personinfo.html?sign=loftuw

JÖNKÖPING UNIVERSITY

School of Engineering

CONTENTS

1. Introduction
   - Visual Programming
   - KNIME
   - Conformal Prediction
   - CP in KNIME

2. Conformal Prediction in KNIME
   - Original content
   - Adding Conformal Regression
   - Addressing Usability Aspects
   - Package structure

3. Use cases
   - Prediction in KNIME
   - Conformal Classification
   - Conformal Regression

4. Evaluation
   - Visualizing Results
   - Experimental setup
   - Results

5. Concluding discussion
   - Cocnlusions
   - Future Work

Introduction

1 ○ ○ ○ ○

Visual Programmin

Machine learning and data science are gradually becoming more and more mainstream.

A particular set of tools offer users with the possibility to build up data science workflows using components.

- Each component represent a specific behaviour, such as data manipulation, modeling or visualization.

There are many tools that offer the possibility to build workflows using visual programming, such as RapidMiner, WEKA, Orange and KNIME.

KNIME

The development of KNIME began in 2004 within a university group and had its first software release in 2006.

KNIME consists of two parts,

- the KNIME Analytics Platform, which is an open source software for end-to-end data analytics using visual programming,
- the KNIME Server, which is an enterprise software for team-based collaboration, automation, management, and deployment of data science workflows.

KNIME has support for most standard machine learning techniques of-the-shelf

A lot of additional functionality is available through community content, providing support for a very wide range of application areas

Conformal Prediction

Conformal prediction has attracted a growing amount of scientific interest in recent years, with a predicted increase in attention onward.

However, it is not yet widely acknowledge outside academia as a natural and important tool to improve the quality of decision support systems relying on predictions.

Making it more easily accessible to a wider public, not necessarily knowledgeable in programming, is important for increased outreach.

Conformal Prediction in KNIME

The Swedish company Redfield AB has previously developed a toolbox for conformal classification for KNIME which has been available through KNIME's official software channels as a community package.

In this tutorial, we introduce an update and extension of the package to include conformal regression as well as a number of additions to improve ease-of-use and accessibility to novice users.

We illustrate the functionality using a number of typical use cases.

Disclaimer: We will assume that the audience has basic familiarity with conformal prediction.

Conformal Prediction in KNIME

○ 2 ○ ○ ○

Original content

Redfield AB deployed the original Conformal Prediction package in KNIME for class conditional conformal classification in 2020.

- Conformal Calibrator - Creates a calibration table with ranking
- Conformal Predictor - Calculates p-values for test or unlabeled data based on the provided calibration table.
- Conformal Classifier - Produces predictions based on the calibration table and the significance level provided by the user.

Original content

- Conformal Scorer - Calculates metrics for estimating conformal predictions.
- Conformal Calibration Loop Start/End - A pair of loop nodes that iterate over the training and calibration data.
- Conformal Prediction Loop Start/End - A pair of loop nodes that iterate over the results from the calibration loop. The Conformal Prediction Loop End node outputs the median of all the iterations.

The Conformal Scorer has been updated to output total aggregated evaluation metrics along with the per-class metrics originally included. The remaining nodes are left as is for compatibility reasons.

Adding Conformal Regression

For the release of the updated Conformal Prediction package, functionality for conformal regression has been added.

The aim was to follow the same logic as used in the original implementation, creating a calibrator, a predictor, a classifier and a scorer node for conformal regression.

However, as opposed to conformal classification where the p-value can be calculated without knowledge of the significance level, the significance level is explicitly used to define the lower and upper bounds of the conformal regressor.

- Consequently, it is not possible to separate the predictor and classifier for regression.

Adding Conformal Regression

The following new nodes have been added to facilitate functionality for conformal regression:

- Conformal Calibrator (Regression) - Calculates the α-values for all calibration instances depending on whether normalization is used or not

- Conformal Predictor and Classifier (Regression) - The node defines the prediction interval using the calibration table, based on the user-defined significance level.

- Conformal Scorer (Regression) - Calculates metrics for estimating conformal regressors.

Addressing Usability Aspects

The most intuitive way to use conformal prediction is to see conformal classification or conformal regression as a single step, taking the calibration and test data together with necessary parameters as input, producing conformal predictions as output, motivating two additional nodes:

- Conformal Classification - The node merges the functionality of the Conformal Calibrator, the Conformal Predictor and the Conformal Classifier into a single node with the sum of the parameters available in the individual nodes.

- Conformal Regression - The node merges the functionality of the Conformal Calibrator (Regression) and the Conformal Predictor and Classifier (Regression) into a single node with the sum of the parameters available in the individual nodes.

Addressing Usability Aspects

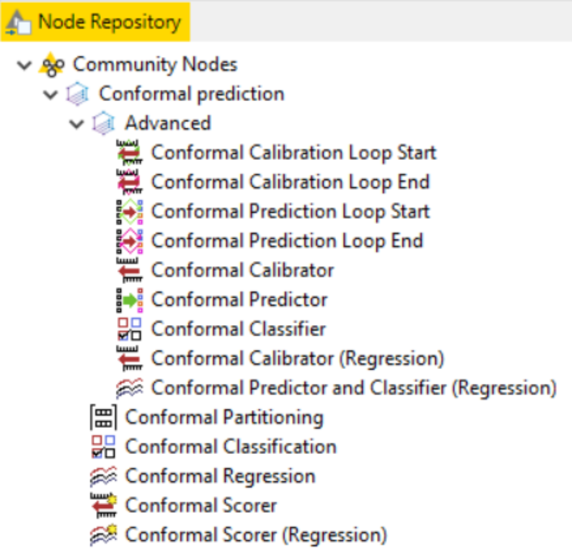
The error rates of the conformal predictors will be exactly the same as the significance levels in the long run but may deviate from the significance level for a single run.

Defining the calibration size to be easily dividable by typical significance levels will reduce the variance in error rates, making each repetition have an error rate closer to the significance level.

To facilitate that, an additional node has been included:

- Conformal Partitioning - A partitioning node adjusting the calibration size for this purpose

Package structure within KNIME



Node Repository

- Community Nodes
  - Conformal prediction
    - Advanced
      - Conformal Calibration Loop Start
      - Conformal Calibration Loop End
      - Conformal Prediction Loop Start
      - Conformal Prediction Loop End
      - Conformal Calibrator
      - Conformal Predictor
      - Conformal Classifier
      - Conformal Calibrator (Regression)
      - Conformal Predictor and Classifier (Regression)
    - Conformal Partitioning
    - Conformal Classification
    - Conformal Regression
    - Conformal Scorer
    - Conformal Scorer (Regression)

Use cases

○ ○ 3 ○ ○

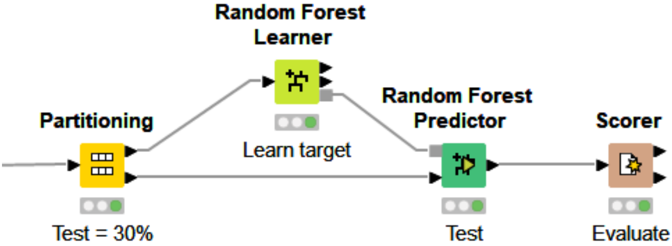
Use cases

To illustrate how conformal prediction can be applied in KNIME, a number of use cases, from simple to more complex, will be presented here.

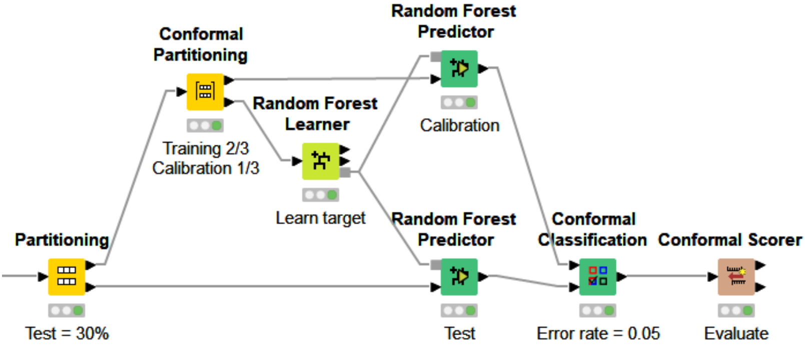
All workflows presented below plus some additional workflows are available for download at KNIME Hub[1] (search for ”COPA 2022” or use the link below).

[1] URL: https://kni.me/s/BuIvXUolHRH3Cy5o

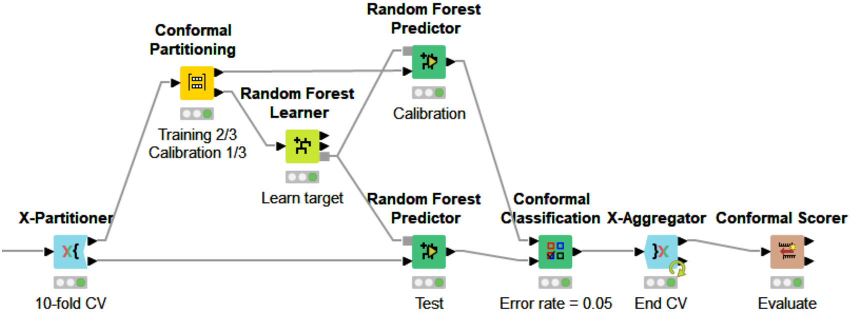
Prediction in KNIME



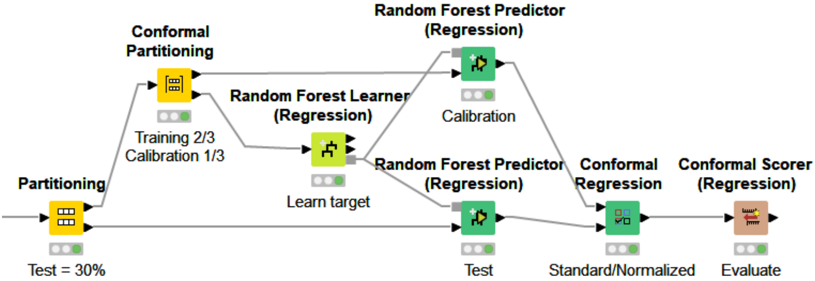
Conformal Classification in KNIME



Conformal Classification using Cross-Validation



Creating a solution for standard conformal regression is analogous to classification.

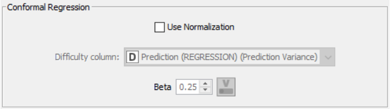


Normalized Conformal Regression

In order to use normalized conformal regression, an additional column of data representing σ, i.e., how difficult an instance is, must exist in the data.

If such a column exist, changing from standard conformal regression to normalized conformal regression requires only opting for using normalization and selecting the σ-column.

Standard Conformal Regression



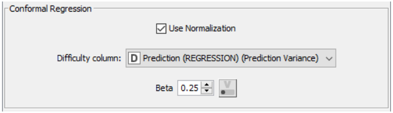
Conformal Regression

☐ Use Normalization

Difficulty column: D Prediction (REGRESSION) (Prediction Variance)

Beta 0.25

Normalized Conformal Regression



Conformal Regression

☑ Use Normalization

Difficulty column: D Prediction (REGRESSION) (Prediction Variance)

Beta 0.25

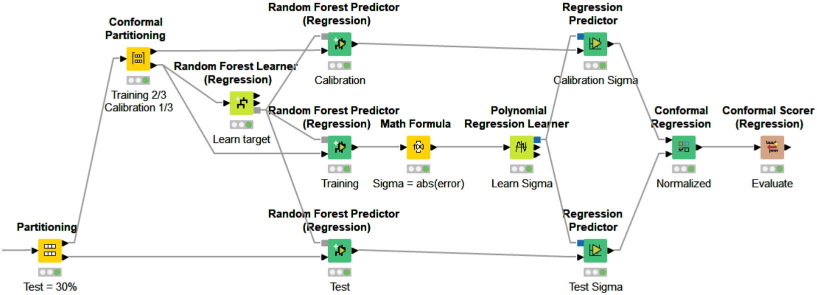
Normalized Conformal Regression

A random forest will automatically add the prediction variance as a separate column along with the prediction.

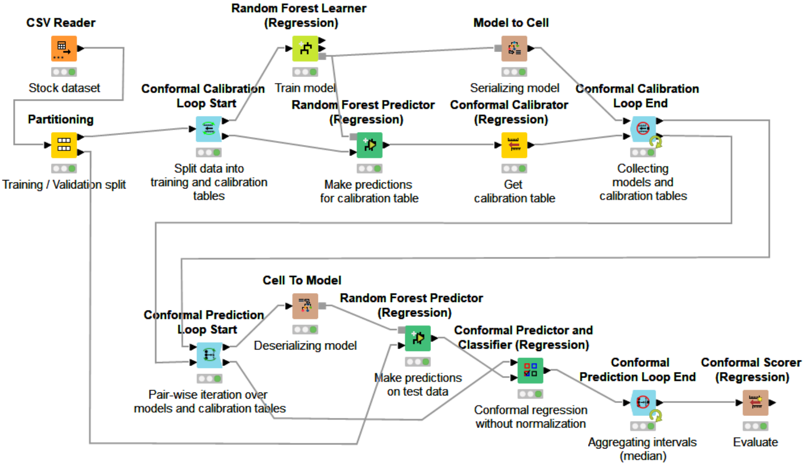
If your underlying model does not automatically provide you with a difficulty estimate, as is the case for most techniques, a difficulty estimation must be defined separately.

One way of doing that is to train a separate model on the absolute error of the first model.

Normalized Conformal Regression



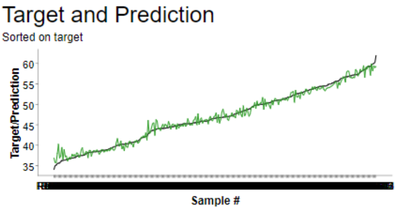
Advanced Conformal Regression using Cross-calibration


CSV Reader
Stock dataset
Partitioning
Training / Validation split
Conformal Calibration Loop Start
Split data into training and calibration tables
Random Forest Learner (Regression)
Train model
Random Forest Predictor (Regression)
Make predictions for calibration table
Model to Cell
Serializing model
Conformal Calibrator (Regression)
Get calibration table
Conformal Calibration Loop End
Collecting models and calibration tables
Conformal Prediction Loop Start
Pair-wise iteration over models and calibration tables
Cell To Model
Deserializing model
Random Forest Predictor (Regression)
Make predictions on test data
Conformal Predictor and Classifier (Regression)
Conformal regression without normalization
Conformal Prediction Loop End
Aggregating intervals (median)
Conformal Scorer (Regression)
Evaluate

Evaluation

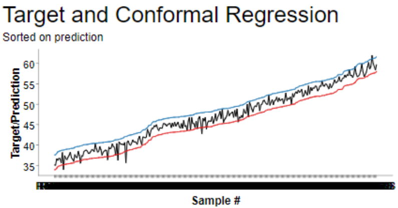
○ ○ ○ 4 ○

Visualizing standard Conformal Regression



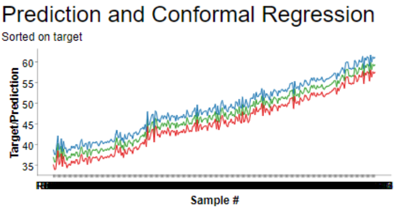
Target and Prediction

Sorted on target



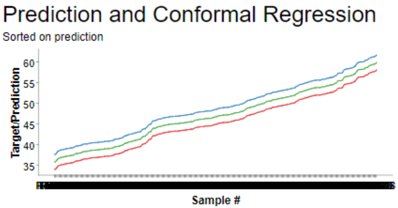
Target and Conformal Regression

Sorted on prediction



Prediction and Conformal Regression

Sorted on target

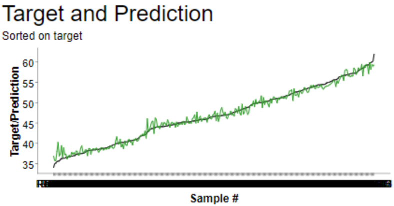


Prediction and Conformal Regression

Sorted on prediction

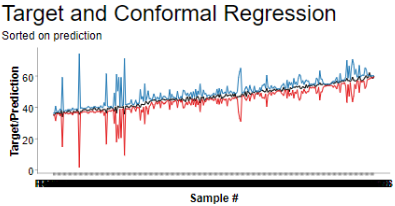
⬛	REGRESSION
🟩	Prediction
🟦	Upper bound
🟥	Lower bound

Visualizing normalized Conformal Regression



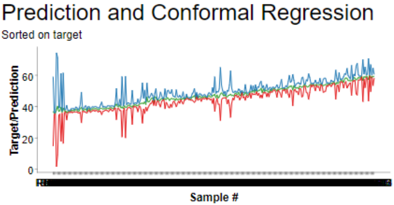
Target and Prediction

Sorted on target



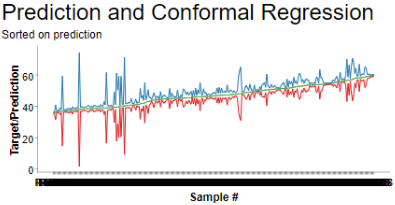
Target and Conformal Regression

Sorted on prediction



Prediction and Conformal Regression

Sorted on target



Prediction and Conformal Regression

Sorted on prediction

Experimental setup

The Conformal Prediction package has also been evaluated in experiments using multiple benchmarking data sets.

25 binary data sets were used for classification and 33 data sets were used for regression.

The tabulated results only include $\epsilon \in \{0.01, 0.05, 0.10\}$.

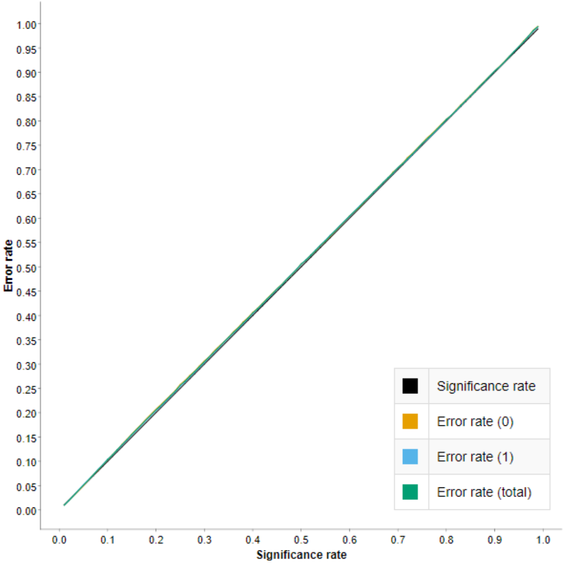
Both experiments are using 10-fold cross validation and the experiments are repeated multiple times.

Regression is evaluated using the three setups presented in previous slides

Classification Experiment

	Error rate			Efficiency		
Data sets	ϵ =0.01	ϵ =0.05	ϵ =0.10	ϵ =0.01	ϵ =0.05	ϵ =0.10
colic	0.01	0.06	0.11	0.16	0.54	0.75
creditA	0.01	0.05	0.10	0.21	0.67	0.92
diabetes	0.01	0.06	0.11	0.11	0.43	0.64
german	0.01	0.05	0.10	0.03	0.16	0.28
haberman	0.01	0.06	0.11	0.04	0.20	0.35
heartC	0.01	0.04	0.09	0.12	0.53	0.75
heartH	0.01	0.05	0.11	0.18	0.60	0.79
heartS	0.01	0.05	0.11	0.11	0.51	0.80
hepati	0.01	0.06	0.11	0.10	0.41	0.68
iono	0.01	0.05	0.10	0.38	0.90	0.95
je4042	0.01	0.05	0.11	0.07	0.36	0.59
je4243	0.01	0.06	0.11	0.05	0.25	0.43
kc1	0.01	0.05	0.10	0.05	0.23	0.39
kc2	0.01	0.07	0.12	0.13	0.49	0.65
kc3	0.01	0.04	0.10	0.04	0.24	0.51
liver	0.01	0.05	0.10	0.05	0.25	0.44
pc1req	0.01	0.04	0.09	0.03	0.17	0.37
pc4	0.01	0.06	0.11	0.27	0.73	0.91
sonar	0.01	0.05	0.10	0.15	0.59	0.75
spect	0.01	0.04	0.09	0.04	0.15	0.29
spectf	0.01	0.05	0.11	0.09	0.42	0.65
transfusion	0.01	0.05	0.09	0.03	0.20	0.34
ttt	0.01	0.04	0.09	0.78	0.97	0.91
vote	0.01	0.05	0.10	0.21	0.62	0.80
wbc	0.01	0.07	0.12	0.40	0.95	0.91
Mean	0.01	0.05	0.10	0.15	0.46	0.63

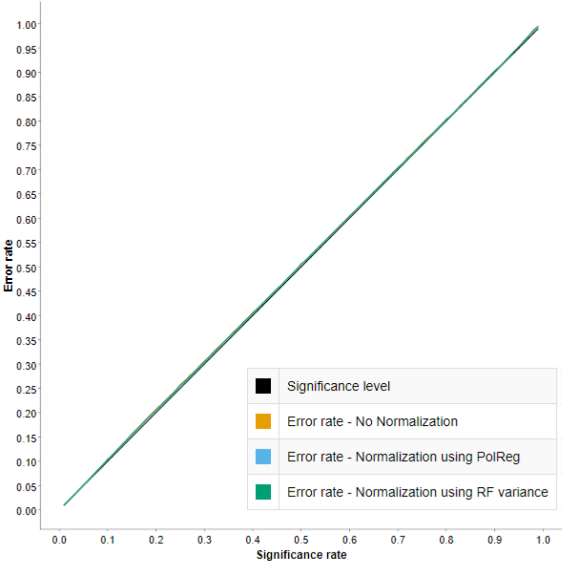
Classification - Error rate vs significance level



Regression - Error rates

	Error rate								
	ϵ =0.01			ϵ =0.05			ϵ =0.10		
Data sets	Std	NPV	NPR	Std	NPV	NPR	Std	NPV	NPR
abalone	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
anacalt	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
bank8fh	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
bank8fm	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
bank8nh	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
bank8nm	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
boston	0.01	0.01	0.01	0.04	0.05	0.05	0.09	0.09	0.08
comp	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
…	…	…	…	…	…	…	…	…	…
laser	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
mg	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
mortage	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
plastic	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
puma8fh	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
puma8fm	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
puma8nh	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
puma8nm	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
quakes	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
stock	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
treasury	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
wineRed	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
wineWhite	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
wizmir	0.01	0.01	0.01	0.05	0.05	0.05	0.10	0.10	0.10
Mean	**0.01**	**0.01**	**0.01**	**0.05**	**0.05**	**0.05**	**0.10**	**0.10**	**0.10**

Regression - Error rate vs significance level



Regression - Efficiency

	Efficiency								
	ϵ =0.01			ϵ =0.05			ϵ =0.10		
Data sets	Std	NPV	NPR	Std	NPV	NPR	Std	NPV	NPR
abalone	0.54	0.52	0.52	0.31	0.30	0.30	0.23	0.22	0.22
anacalt	0.60	0.45	0.54	0.25	0.23	0.23	0.16	0.15	0.15
bank8fh	0.55	0.54	0.53	0.39	0.38	0.38	0.31	0.30	0.30
bank8fm	0.33	0.32	0.31	0.20	0.19	0.19	0.15	0.15	0.15
bank8nh	0.79	0.78	0.77	0.45	0.44	0.44	0.33	0.32	0.32
bank8nm	0.43	0.40	0.40	0.23	0.22	0.22	0.16	0.15	0.15
boston	0.77	0.62	0.73	0.33	0.31	0.33	0.23	0.22	0.23
comp	0.20	0.19	0.19	0.12	0.12	0.12	0.09	0.09	0.09
…	…	…	…	…	…	…	…	…	…
laser	0.34	0.30	0.33	0.10	0.09	0.10	0.06	0.05	0.05
mg	0.67	0.61	0.66	0.40	0.37	0.40	0.28	0.26	0.28
mortage	0.09	0.09	0.09	0.05	0.05	0.05	0.03	0.03	0.03
plastic	0.85	0.86	0.85	0.65	0.65	0.65	0.54	0.54	0.54
puma8fh	0.76	0.75	0.75	0.57	0.56	0.57	0.48	0.47	0.47
puma8fm	0.36	0.36	0.36	0.27	0.27	0.27	0.22	0.22	0.22
puma8nh	0.75	0.73	0.74	0.54	0.53	0.54	0.44	0.43	0.44
puma8nm	0.39	0.37	0.38	0.28	0.27	0.28	0.23	0.23	0.23
quakes	1.06	1.08	1.06	0.71	0.70	0.71	0.55	0.54	0.55
stock	0.18	0.18	0.18	0.11	0.11	0.11	0.09	0.09	0.09
treasury	0.12	0.12	0.12	0.06	0.06	0.06	0.03	0.03	0.03
wineRed	0.74	0.72	0.73	0.50	0.49	0.50	0.38	0.38	0.38
wineWhite	0.66	0.65	0.66	0.44	0.43	0.43	0.34	0.33	0.34
wizmir	0.15	0.15	0.15	0.08	0.08	0.08	0.07	0.07	0.07
Mean	**0.48**	**0.47**	**0.47**	**0.31**	**0.30**	**0.30**	**0.24**	**0.23**	**0.24**

Concluding discussion

○ ○ ○ ○ 5

Conclusions

In this tutorial, the extended Conformal Prediction package in KNIME has been introduced and explained using a number of straight-forward use cases and experiments.

The package offer the benefits of conformal prediction in an accessible and easy-to-use way while still providing advanced options for more demanding needs.

Future Work

Future planned expansions of the package include adding support for Venn and Venn-Abers predictors for classification and standard and normalized conformal predictive systems for regression.

Several other smaller or larger updates are considered, such as:

- Mondrian categories
- Smoothed p-values
- Interpolation
- Enriching interpretable models with confidence estimates

AVAILABLE RESOURCES

The updated package will be released with KNIME 4.7 in about a month.

If you want early access (Warning - stability not fully guaranteed):

1. Go to File -> Preferences -> Install/Update -> Available Update Sites -> Add...
2. Add https://update.knime.com/community-contributions/trunk
3. Install the Conformal Prediction package version 2.0

All workflows used in paper are available at KNIME Hub
(URL: https://kni.me/s/BuIvXUolHRH3Cy5o or search for ”COPA 2022”).

Additional workflows can be found at Redfields KNIME Hub:
https://kni.me/w/anURi0RlGm7pm4xZ