DNB UNRESTRICTED

Discovering and ranking validation rules in supervisory data

Willem Jan Willemse and Annick van Ool

Central bankers go data driven: applications of AI and ML for policy and prudential supervision conference, 13th May 2022

DeNederlandscheBank
EUROSYSTEEM

Agenda

DNB UNRESTRICTED

- **Motivation**
- Rulemining
- Applications of rulemining
- Conclusions and extensions

Motivation

DNB UNRESTRICTED

- **Data-driven supervision increases data quality requirements**
- Current reporting frameworks have some limitations
- New techniques & tools needed to assess data quality of granular data

- **We introduce the ruleminer software package to**
  - automatically generate large sets of validation rules
  - discover underlying data patterns in datasets, and

- **We then rank the validation rules with a statistical machine learning model**

Related literature

- Rules mining is subset of the field of data mining
- Association rules mining:
  - Categorical: *e.g. Agrawal et al. (1993), Agrawal and Srikant (1994)*
  - Quantitative: *e.g. Piatetsky-Shapiro (1991), Fukuda et al. (1996), Salleb-Aouissi et al. (2013), Jaramillo et al. (2021)*
- Rules ranking:
  - Optimization approach using data envelopment analysis: *Charnes et al. (1978))*
  - Linear regression including expert ranking & quantitative metrics: *Chen (2007), Bazaldua et al (2014), Toloo et al. (2019)*

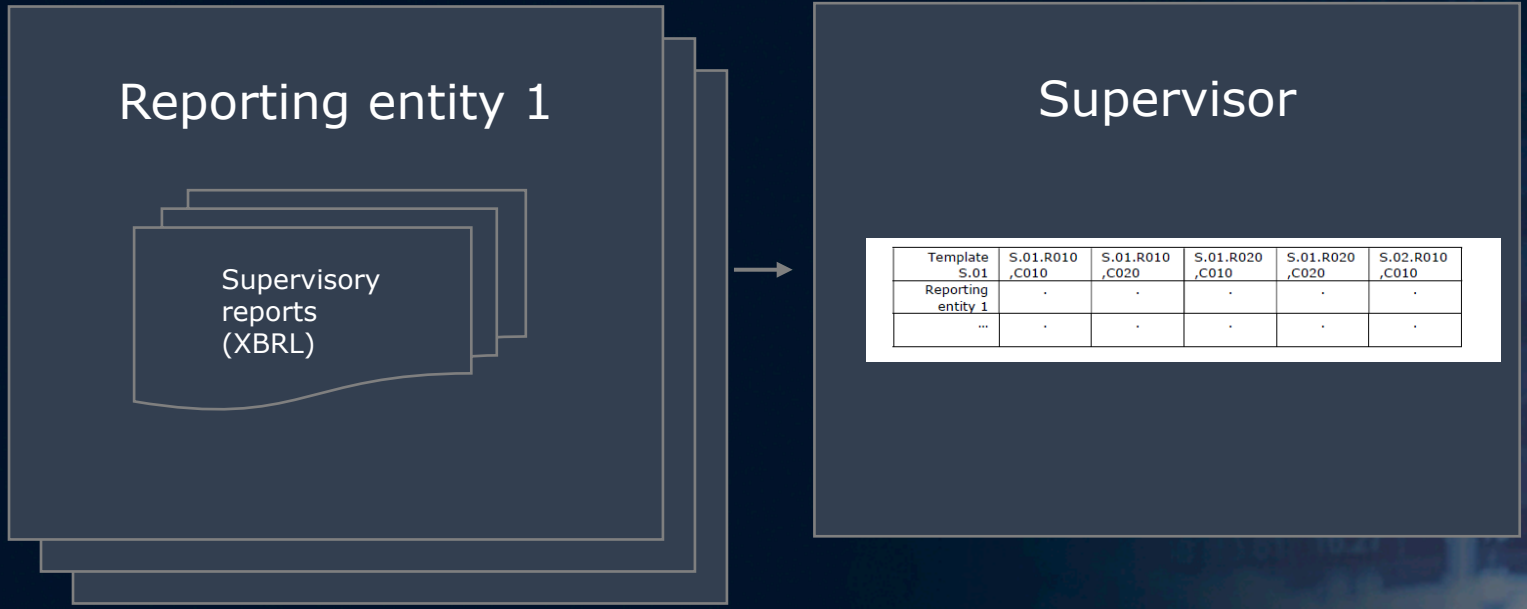
Agenda

DNB UNRESTRICTED

- Motivation
- **Rulemining**
- Applications
- Conclusions and extensions

Rulemining: supervisory data process

DNB UNRESTRICTED



Financial entities submit their supervisory reports quarterly and annually to the national supervisor

Rulemining: limitations current reporting framework

DNB UNRESTRICTED

- Very large sets of validation rules (>10.000) not feasible
- Validation rules between two subsequent supervisory reports not possible
- Using external data sources in validation rules not possible

Our approach to rulemining: the rule grammar

- Two rule categories:
  1. Conditional or transactional rules: *'IF X THEN Y'*
  2. Quantitative rules: e.g. *'X = Y'*
- We formulated a rule grammar to formulate a wide range of possible **rule expressions**
- **Regular expressions** can be used for strings and column names within rule expressions

EXAMPLE	RULE
1	{"A"} ≥ 0
2	{"A"} + {"B"} = {"C"}
3	IF {"A"} = "reported" THEN {"B"} = 0

EXAMPLE	REGEX	RESULT
1	.*	S.01,R010,C010, S.01,R010,C020, S.01,R020,C010, S.01,R020,C020, S.02,R010,C010
2	S.01,R010,C010	S.01,R010,C010
3	S.01,R010,C\d+	S.01,R010,C010, S.01,R010,C020

Our approach to rulemining: rule discovery & rule metrics

- Rulemining algorithm generates all possible rules given rule expression
- Pruning algorithm removes semantically identical rules
- Various association rules metrics are calculated for all possible rules
  - support
  - confidence
  - lift
  - added value
  - rule power factor
  - conviction
  - casual confidence

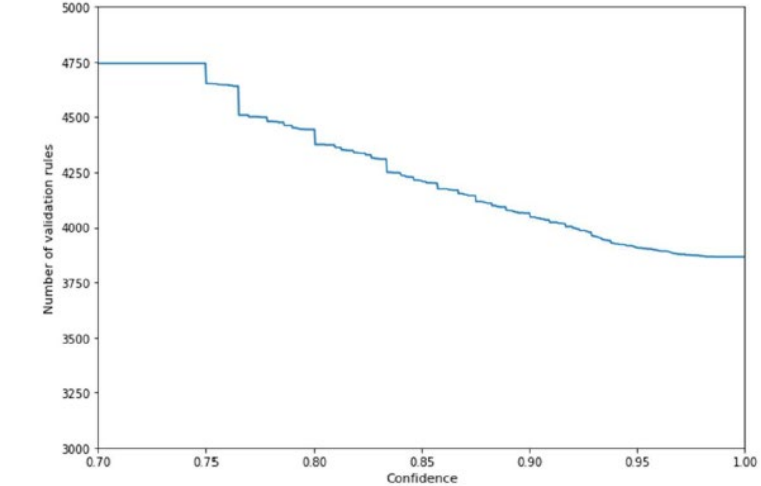
→

1. (({"4"}>{"3"}) & (({"2"}+{"1"})={"0"}))
2. ((({"1"}+{"2"})={"0"}) & ({"4"}>{"3"}))

METRIC	DEFINITION	INTERPRETATION
absolute support count	n_X	The support count is the number of occurrences of a pattern in a dataset, i.e. the number of transactions in the database that satisfy or confirm the pattern
absolute exception count	$n - n_X$	The exception count is the number of transactions in the database that do not satisfy the pattern
support	$supp(X) = {}^{n_X}/_{n}$	Support is an indication of how frequently the pattern appears in the dataset. It is defined as the proportion of transactions in the dataset which contains the given pattern. Support is often used to represent the significance of a pattern
confidence	$conf(X \rightarrow Y) = {}^{n_{XY}}/_{n_X}$	Confidence is an indication of how often a rule has been found to be true. Thus

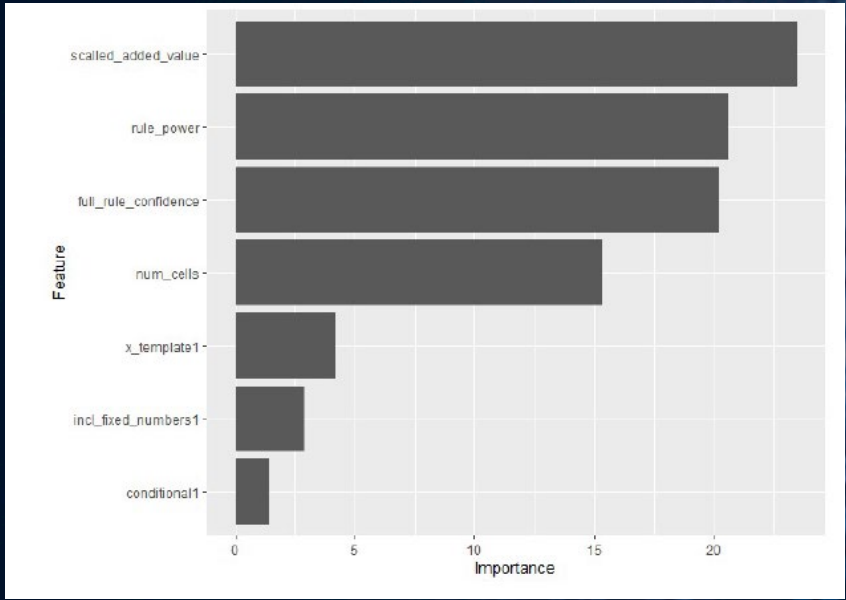
Our approach to rulemining: rule constraints

- Amount of generated validation rules depends on constraints imposed on certain metrics



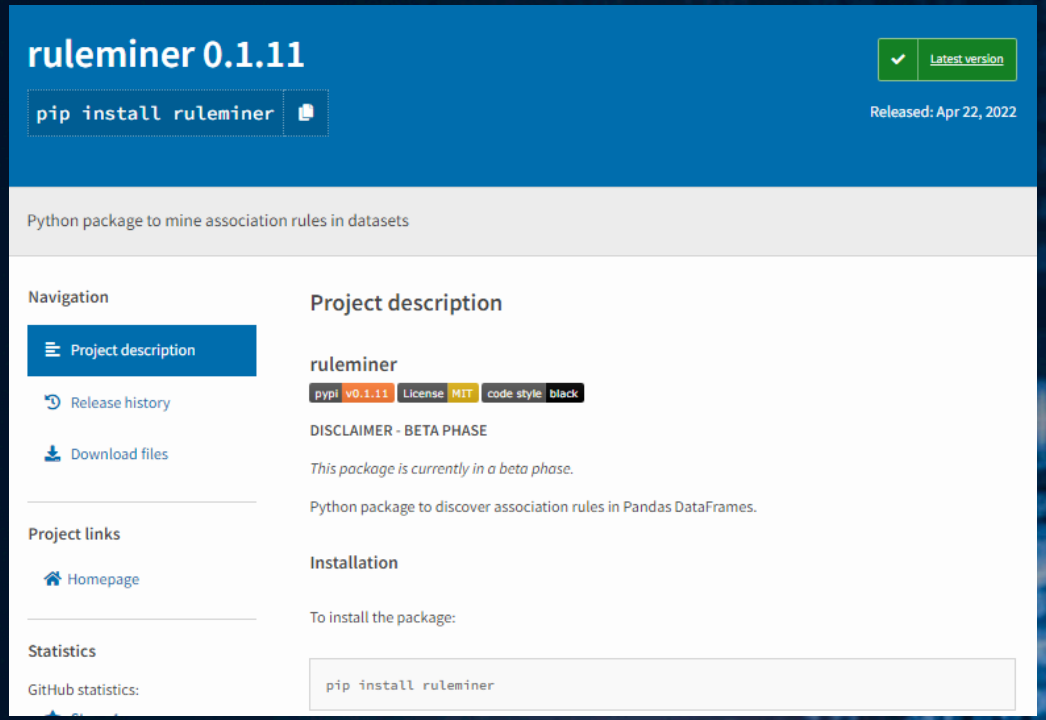
Our approach to rulemining: rule ranking

- Dataset with generated validation rules including quantitative properties, statistical metrics and expert label (important/not important)
- Trained supervised ML algorithm (random forest) → output: rating (probability) of the rule importance between 0 and 1
- Accuracy 85%



Ruleminer software package

The Ruleminer software package is an open source Python package and available on pypi.org



Agenda

DNB UNRESTRICTED

- Motivation
- Rulemining
- **Applications**
- Conclusions and extensions

Applications

DNB UNRESTRICTED

1. Granular asset data
2. **Actuarial run-off triangles between two periods**
3. Solvency 2 data
4. **Publishing validation rules**

Application: actuarial run-off triangles between two periods

Reported run-off triangle t

Accident year	Development year				
	n-4 (c010)	n-3 (c020)	n-2 (c030)	n-1 (c040)	n (c050)
n-4 (r010)	60	29	15	6	3
n-3 (r020)	24	34	21	8	
n-2 (r030)	32	16	11		
n-1 (r040)	19	27			
n (r050)	40				

Reported run-off triangle t+1

Accident year	Development year				
	n-4 (c010)	n-3 (c020)	n-2 (c030)	n-1 (c040)	n (c050)
n-4 (r010)	24	34	21	8	4
n-3 (r020)	32	16	11	9	
n-2 (r030)	19	27	16		
n-1 (r040)	40	19			
n (r050)	56				

Application: actuarial run-off triangles between two periods (2)

Rule template:

if ({"column1 (t)"} != 0)& ({"column2 (t+1)"} != 0) then ({"column1 (t)"} = {"column2 (t+1)"})

Example generated rule:

if ({"r050c010 (t)"} != 0)& ({"r040c010 (t+1)"} != 0) then ({"r050c010 (t)"} = {"r040c010 (t+1)"})

Number of validation rules is high for insurance undertakings with several lines of business:
11.450 validation rules in case of 12 lines of business

```
params = {'metrics': ['abs support', 'abs exceptions', 'confidence'],
          'filter': {'abs support': 5, 'confidence': 0.3},
          'decimal': -3}
templates = []
#Define templates
for column1 in columns:
    for column2 in columns:
        expr = 'if (({"' + column1 + ' \(t-1\)"} != 0)&({"' + column2 + ' \(t\)"} != 0)) then ({"' + column1 + ' \(t-1\)"}
        templates = templates + [{'expression': expr}]
r = ruleminer.RuleMiner(templates=templates, data=dfa, par
```

```
]: 1 r = ruleminer.RuleMiner(data=dfa,rules=r.rules)
   2 r.evaluate()
```

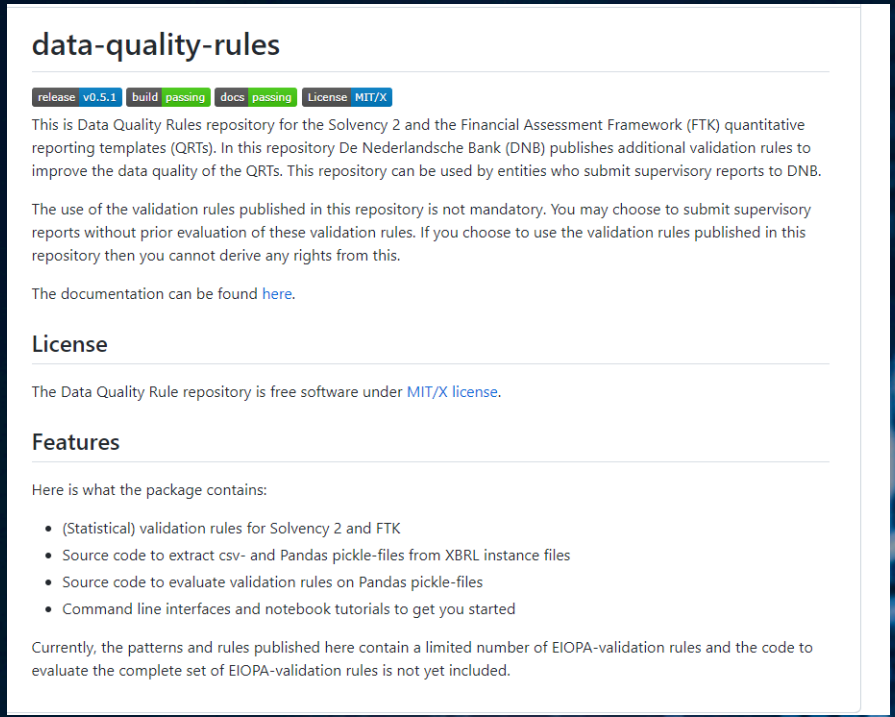
2022-05-09 13:41:52,828 Finished

]: Export this dataframe (630 rows, 9 cols)

	rule_id	rule_group	rule_definition	rule_status	abs support	abs exceptions	confidence	result	indices
0	0	0	if(({"S2ARS.S.19.01.01.01,r0140,c0010 (t-1)"}!...		603	27	0.957143	True	Other motor insurance [direct business...
1	0	0	if(({"S2ARS.S.19.01.01.01,r0140,c0010 (t-1)"}!...		603	27	0.957143	True	Fire and other damage to property insu...
			if(({"S2ARS.S.19.01.01.01,r0140,c0010 (t-						General liability insurance

Application: publishing validation rules

- Generated validation rules can be published and shared with financial entities.
- This allows financial entities to check the validation rules before they submit the report to the supervisor.
- We did two pilot with several Dutch insurance undertakings.



data-quality-rules

release v0.5.1 | build passing | docs passing | License MIT/X

This is Data Quality Rules repository for the Solvency 2 and the Financial Assessment Framework (FTK) quantitative reporting templates (QRTs). In this repository De Nederlandsche Bank (DNB) publishes additional validation rules to improve the data quality of the QRTs. This repository can be used by entities who submit supervisory reports to DNB.

The use of the validation rules published in this repository is not mandatory. You may choose to submit supervisory reports without prior evaluation of these validation rules. If you choose to use the validation rules published in this repository then you cannot derive any rights from this.

The documentation can be found here.

License

The Data Quality Rule repository is free software under MIT/X license.

Features

Here is what the package contains:

- (Statistical) validation rules for Solvency 2 and FTK
- Source code to extract csv- and Pandas pickle-files from XBRL instance files
- Source code to evaluate validation rules on Pandas pickle-files
- Command line interfaces and notebook tutorials to get you started

Currently, the patterns and rules published here contain a limited number of EIOPA-validation rules and the code to evaluate the complete set of EIOPA-validation rules is not yet included.

Agenda

DNB UNRESTRICTED

- Motivation
- Rulemining
- Applications
- **Conclusions and extensions**

Conclusion and extensions

- **Ruleminer** enables discovery of wide range of rules that goes beyond what is feasible and possible within current reporting frameworks
- Relevant rules are selected using rule metrics and a supervised ML algorithm
- Application to other datasets, e.g. loan level data

- Several extensions are possible, for example
  - discover time-dependent rules and / or entity-specific rules;
  - strengthen rules with outcome of outlier detection models and, vice versa, explain outliers with transparant rules