Formal Description of Validity Criteria for Data and Automated Cloud-Based Validity Checking to Generate a Validated Data Set

Lucas Krauße
Advisor: Christian Vetter
Chair of Decentralized Systems Engineering
https://dse.in.tum.de/
In cooperation with BMW Group

15.02.2022 – 12.08.2022

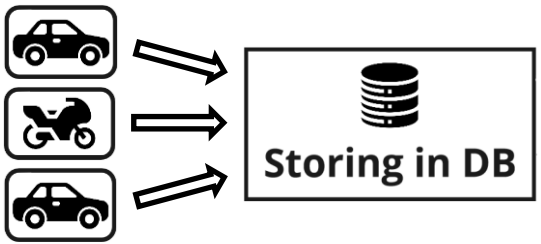
Motivation

- BMW processes data from customer vehicles that statistically describe driving behavior
  => **FASTA** data

Motivation

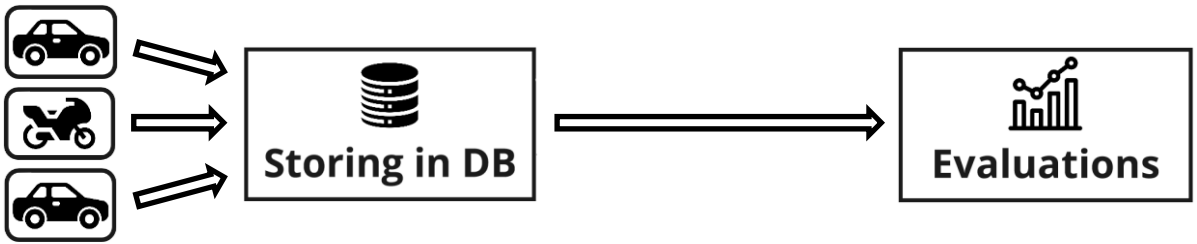
- BMW processes data from customer vehicles that statistically describe driving behavior
  => **FASTA** data

- Every day there are approximately 100,000 new vehicle readouts which are stored but **not validated**



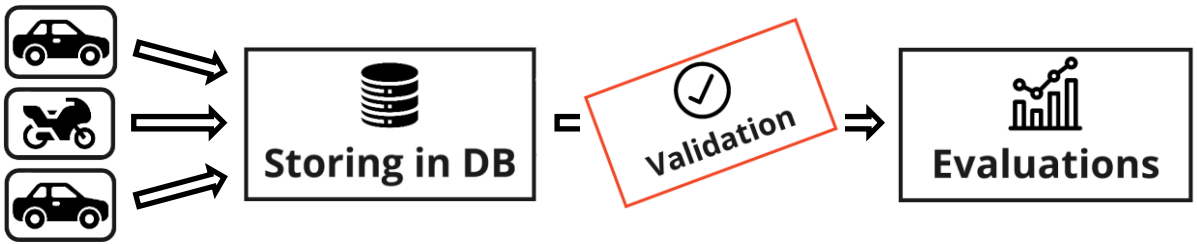
Motivation

- BMW processes data from customer vehicles that statistically describe driving behavior
  => **FASTA** data
- Every day there are approximately 100,000 new vehicle readouts which are stored but **not validated**
- Based on FASTA data, evaluations are made for a wide variety of areas



Motivation

- BMW processes data from customer vehicles that statistically describe driving behavior
  => **FASTA** data
- Every day there are approximately 100,000 new vehicle readouts which are stored but **not validated**
- Based on FASTA data, evaluations are made for a wide variety of areas
- Due to measurement errors, sensor failures or faulty software these can be **incorrect**



Outline

- ~~Motivation~~
- Problem Statement
- Data Validation Requirements
- Pandera (Data Validation Library)
- Conditional Language
- Validation of Multidimensional Values
- Evaluation
- Conclusion

Problem Statement

- Formal validation of FASTA data:

Problem Statement

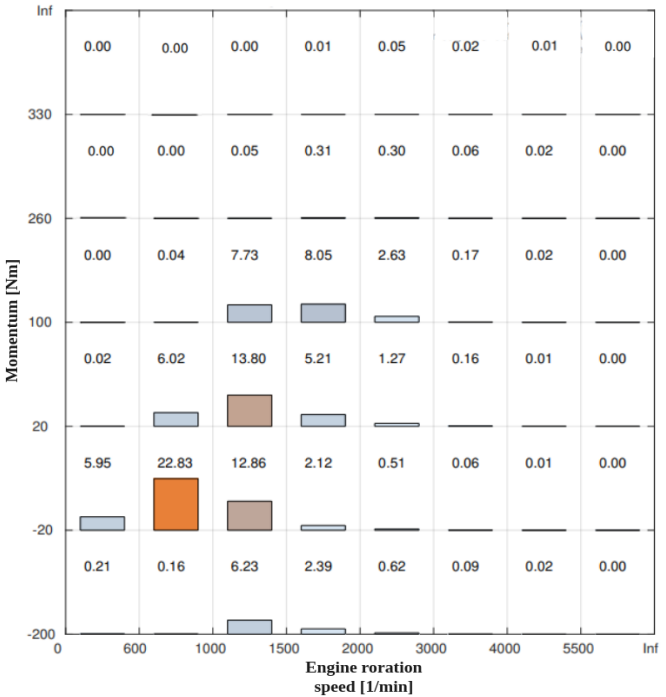
- Formal validation of FASTA data:
  - Individual columns (primitive data types)

VNR	Power	Km_Per_Day	Series_Production
MK41480	141	49.9507389	1
SR57464	190	30.7980456	0

Example individual columns

Problem Statement

- Formal validation of FASTA data:
  - Individual columns (primitive data types)
  - Multidimensional values (e.g. Histograms)



Example: KF-PROZ multidimensional values

Problem Statement

- Formal validation of FASTA data:
  - Individual columns (primitive data types)
  - Multidimensional values (e.g. Histograms)
  - Combination of individual attributes

Problem Statement

- Formal validation of FASTA data:
  - Individual columns (primitive data types)
  - Multidimensional values (e.g. Histograms)
  - Combination of individual attributes
  - Conditional rules

OperatingDays	ReadoutDate	ProductionDate	RegisteredDate
0	736982	736961	
921	737903	736961	736982

Example: Conditional rules and combination of individual attributes

Problem Statement

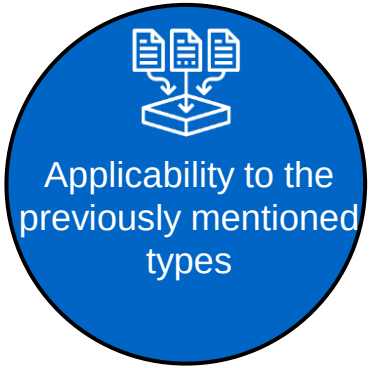
- Formal validation of FASTA data:
  - Individual columns (primitive data types)
  - Multidimensional values (e.g. Histograms)
  - Combination of individual attributes
  - Conditional rules

- Pipeline:

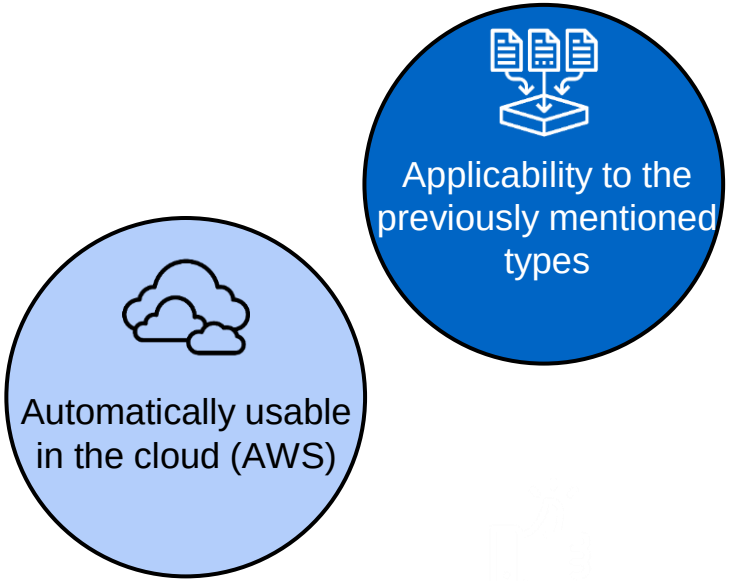
Outline

- ~~Motivation~~
- ~~Problem Statement~~
- Data Validation Requirements
- Pandera (Data Validation Library)
- Conditional Language
- Validation of Multidimensional Values
- Evaluation
- Conclusion

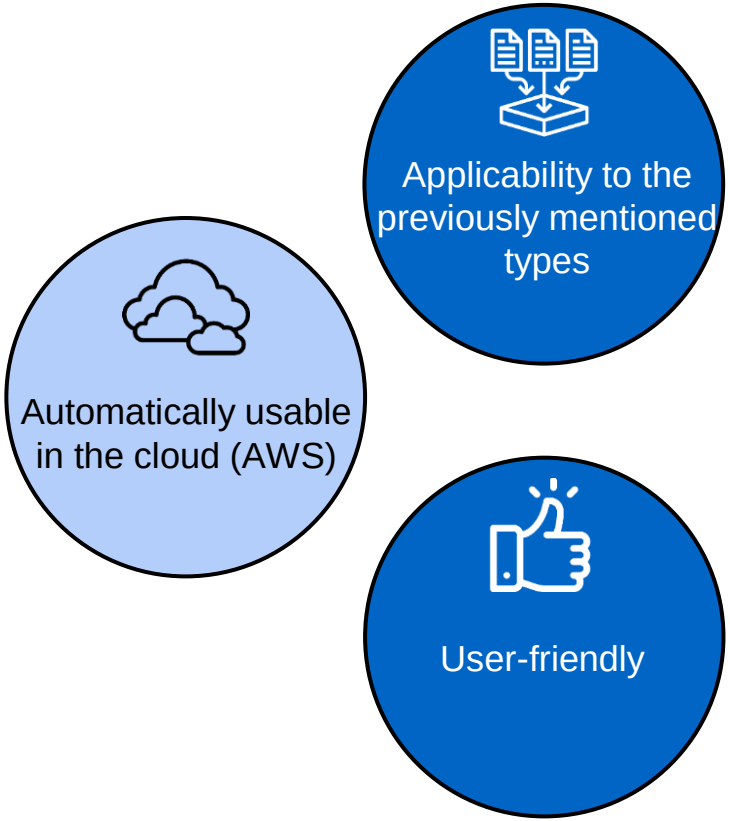
Data Validation Requirements



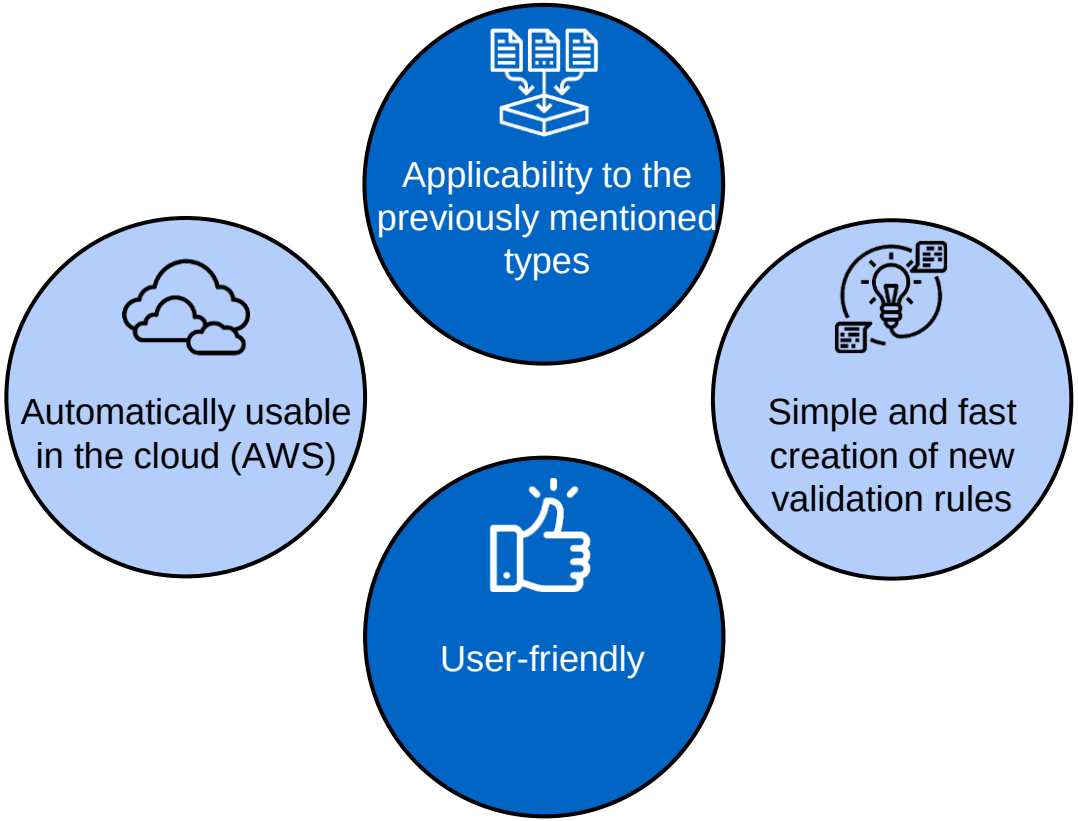
Data Validation Requirements



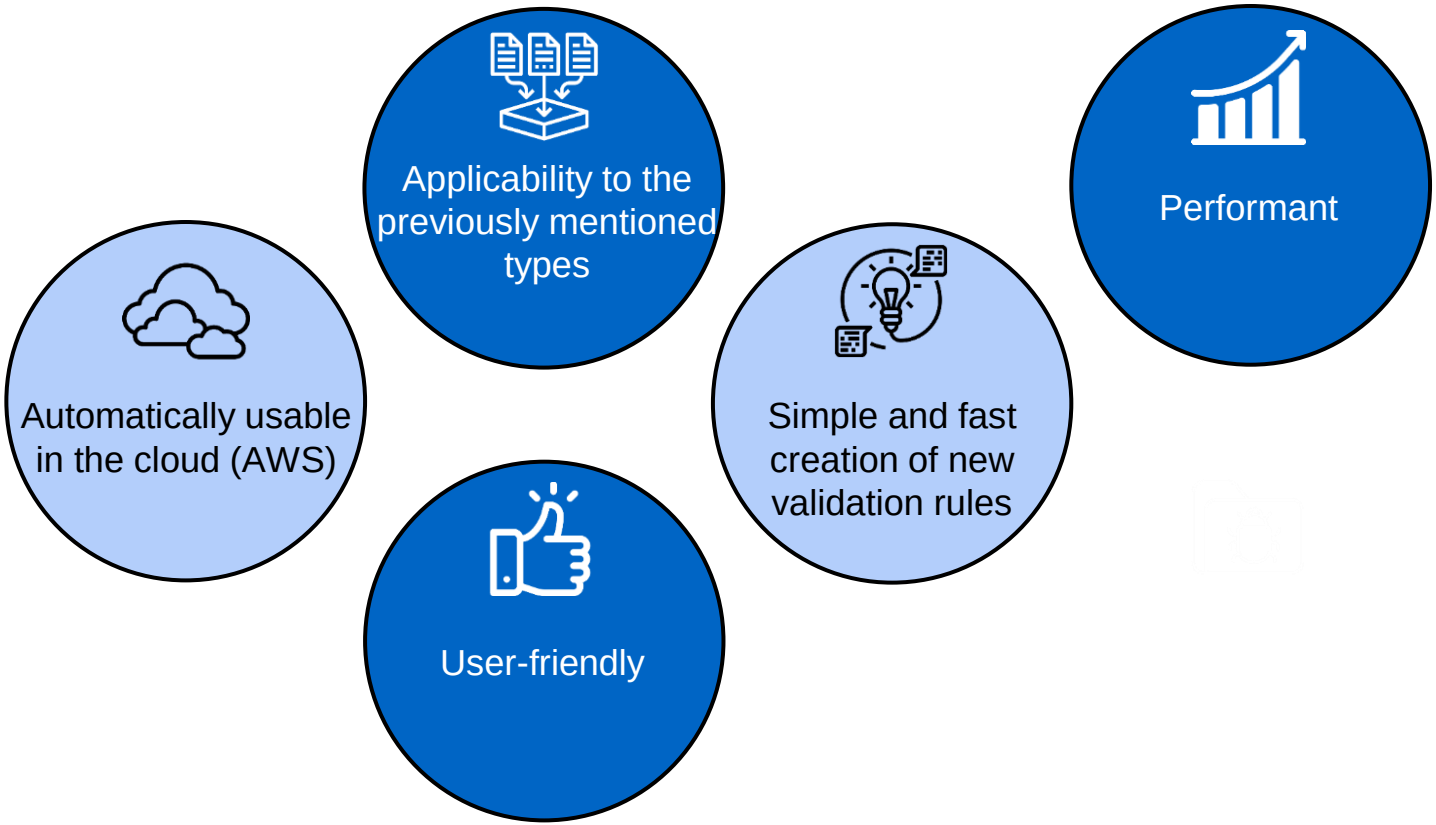
Data Validation Requirements



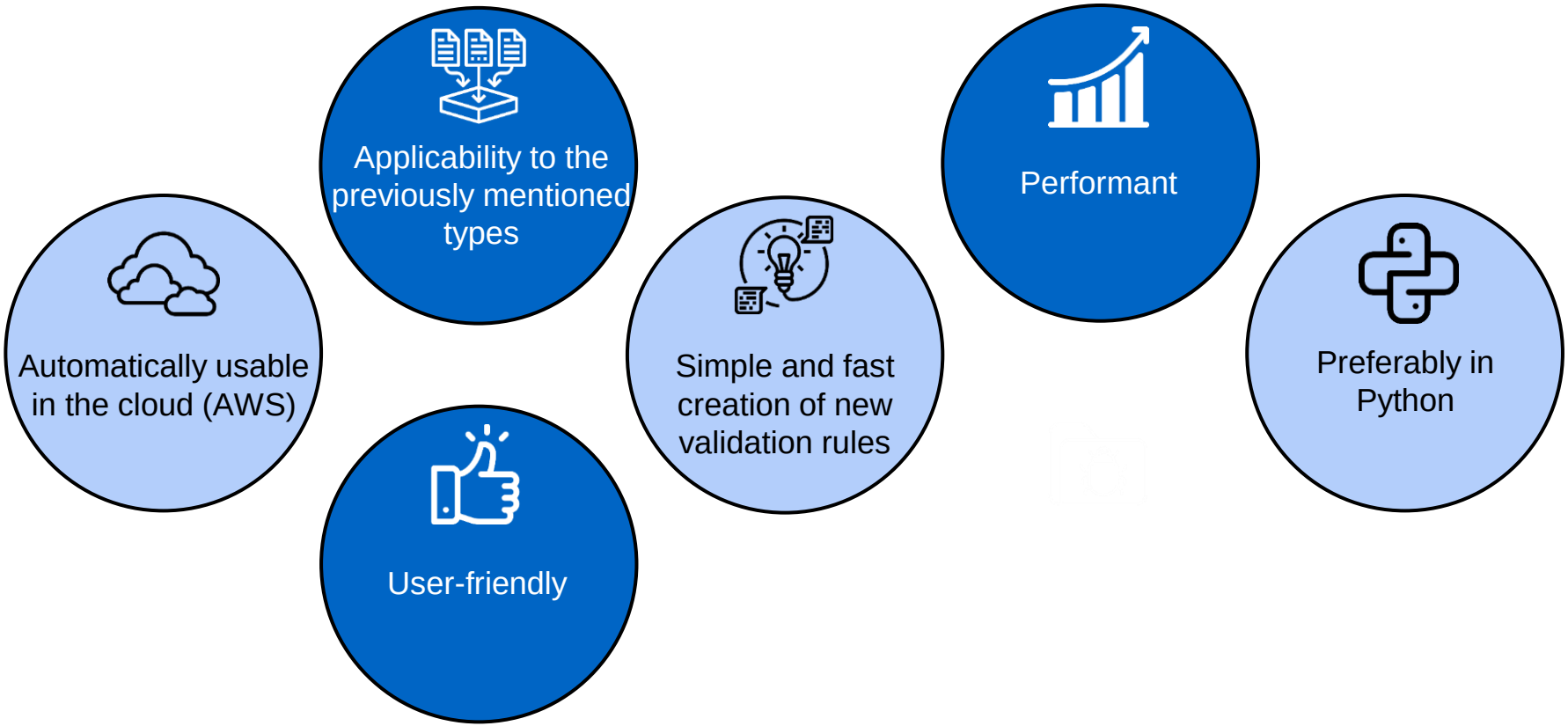
Data Validation Requirements



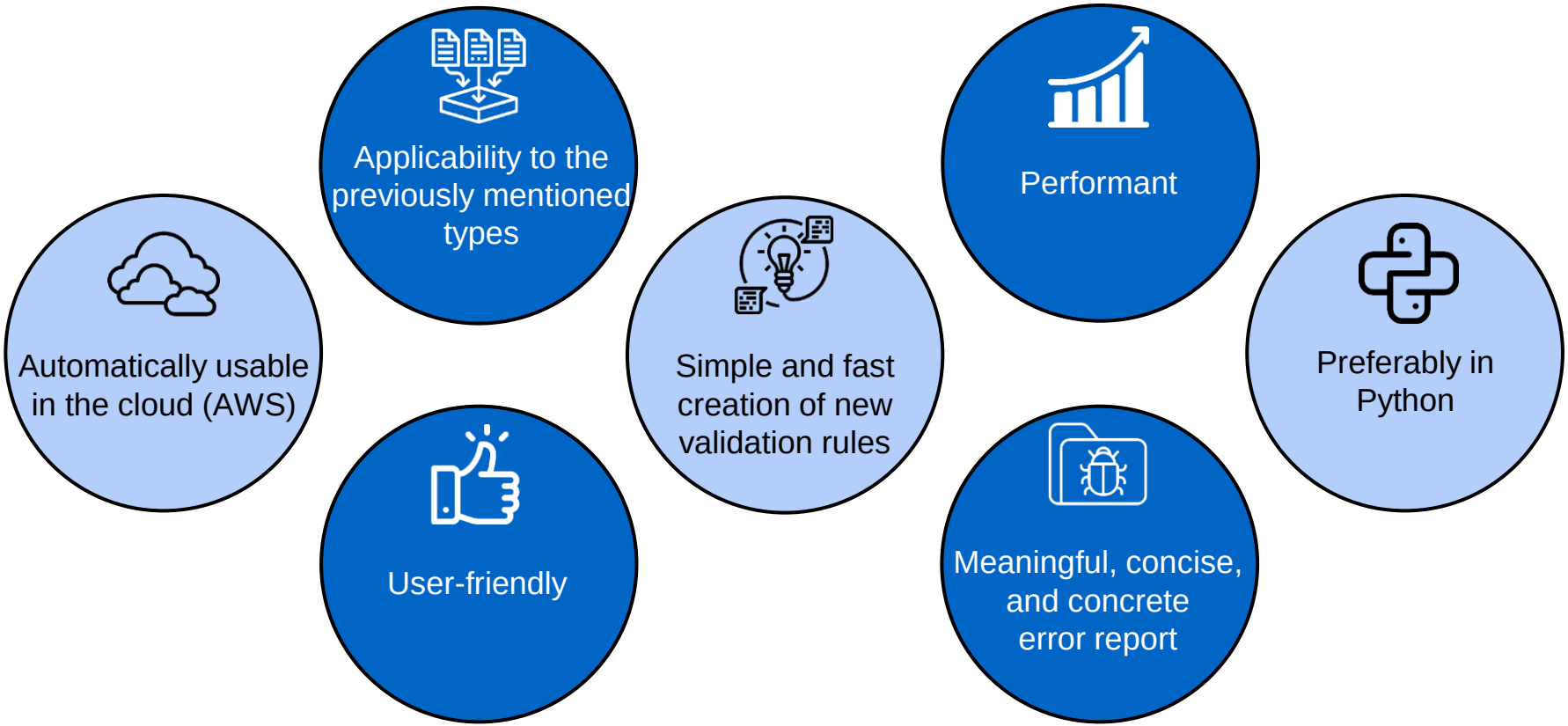
Data Validation Requirements



Data Validation Requirements



Data Validation Requirements

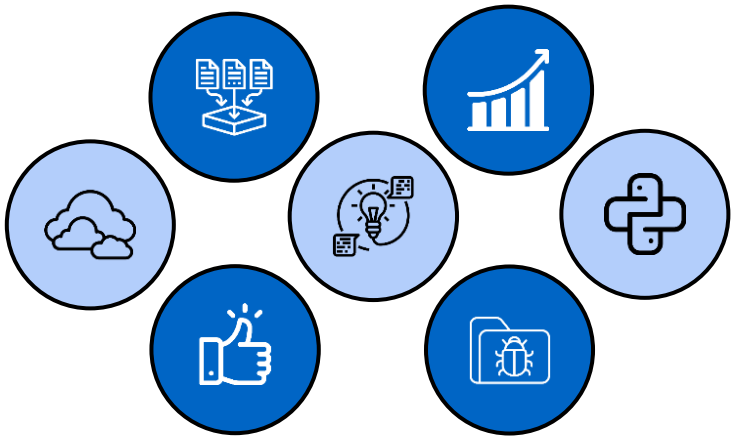


Outline

- ~~Motivation~~
- ~~Problem Statement~~
- ~~Data Validation Requirements~~
- Pandera (Data Validation Library)
- Conditional Language
- Validation of Multidimensional Values
- Evaluation
- Conclusion

Pandera [1]

- Definition of a Schema
- At runtime data validation on DataFrame (Excel table)
- Adding own (more complex) validation rules by an annotation
- Error report as DataFrame (Excel tables)



index	check	column	schema_context	check_number	failure_case
0	OperatingDays_validation	---	DataFrameSchema	---	---
	TMOT_temp_order	TMOT	Column	2	[63.2755 28.75814 0 0 0]
	sum_up_to_100	TMOT	Column	1	[63.2755 28.75814 0 0 0]
21	not_nullable	AvgTripLen	Column		
48	date_before_current_date	ReadoutDate	Column	1	738293
1050	in_range(0, 1000)	AvgTripLen	Column	0	1601,285714
1582	greater_than_or_equal_to(0)	OperatingDays	Column	0	-17

[1] https://pandera.readthedocs.io/en/stable/

Outline

- ~~Motivation~~
- ~~Problem Statement~~
- ~~Data Validation Requirements~~
- ~~Pandera (Data Validation Library)~~
- Conditional Language
- Validation of Multidimensional Values
- Evaluation
- Conclusion

Pandera Extended by Conditional Language

- Creation of more concise validation rules

Pandera Extended by Conditional Language

- Creation of more concise validation rules
- DSL (=Domain Specific Language)

Pandera Extended by Conditional Language

- Creation of more concise validation rules
- DSL (=Domain Specific Language)
- Allowed literals are **primitive data types, column names, lists, null checks** and pandas **DataFrames/Series**

Pandera Extended by Conditional Language

- Creation of more concise validation rules
- DSL (=Domain Specific Language)
- Allowed literals are **primitive data types, column names, lists, null checks** and pandas **DataFrames/Series**
- Typical **comparison operators, negation** and **in**-list operator

Pandera Extended by Conditional Language

- Creation of more concise validation rules
- DSL (=Domain Specific Language)
- Allowed literals are **primitive data types, column names, lists, null checks** and pandas **DataFrames/Series**
- Typical **comparison operators, negation** and **in**-list operator
- **Addition, subtraction,** (integer) **multiplication,** and (integer) **division**

Pandera Extended by Conditional Language

- Creation of more concise validation rules
- DSL (=Domain Specific Language)
- Allowed literals are **primitive data types, column names, lists, null checks** and pandas **DataFrames/Series**
- Typical **comparison operators, negation** and **in**-list operator
- **Addition, subtraction,** (integer) **multiplication,** and (integer) **division**
- Logical operators: **AND** and **OR**

Pandera Extended by Conditional Language

- Creation of more concise validation rules
- DSL (=Domain Specific Language)
- Allowed literals are **primitive data types, column names, lists, null checks** and pandas **DataFrames/Series**
- Typical **comparison operators, negation** and **in**-list operator
- **Addition, subtraction,** (integer) **multiplication,** and (integer) **division**
- Logical operators: **AND** and **OR**

OperatingDays	ReadoutDate	ProductionDate	RegisteredDate
0	736982	736961	
921	737903	736961	736982

Example: Conditional rules and combination of individual attributes

- `"NOT RegisteredDate.isnan AND OperatingDays == ReadoutDate – RegisteredDate OR RegisteredDate.isnan AND OperatingDays == ReadoutDate - ProductionDate"`

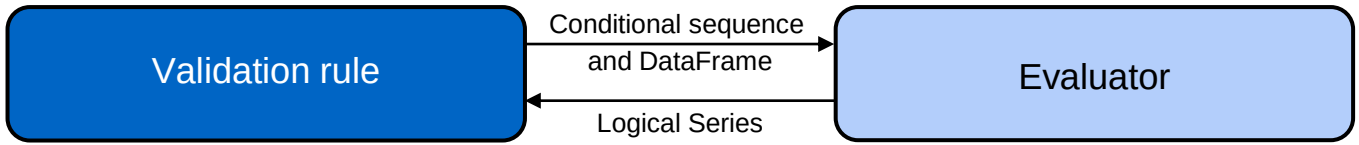
Pandera Extended by Conditional Language

- Creation of more concise validation rules
- DSL (=Domain Specific Language)
- Allowed literals are **primitive data types, column names, lists, null checks** and pandas **DataFrames/Series**
- Typical **comparison operators, negation** and **in**-list operator
- **Addition, subtraction,** (integer) **multiplication,** and (integer) **division**
- Logical operators: **AND** and **OR**

OperatingDays	ReadoutDate	ProductionDate	RegisteredDate
0	736982	736961	
921	737903	736961	736982

Example: Conditional rules and combination of individual attributes

- `"NOT RegisteredDate.isnan AND OperatingDays == ReadoutDate – RegisteredDate OR RegisteredDate.isnan AND OperatingDays == ReadoutDate - ProductionDate"`

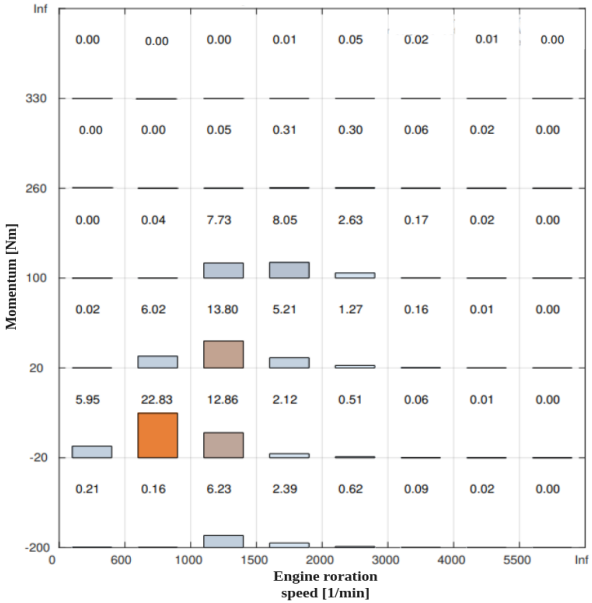


Outline

- ~~Motivation~~
- ~~Problem Statement~~
- ~~Data Validation Requirements~~
- ~~Pandera (Data Validation Library)~~
- ~~Conditional Language~~
- Validation of Multidimensional Values
- Evaluation
- Conclusion

Validation of Multidimensional Values – KF-PROZ

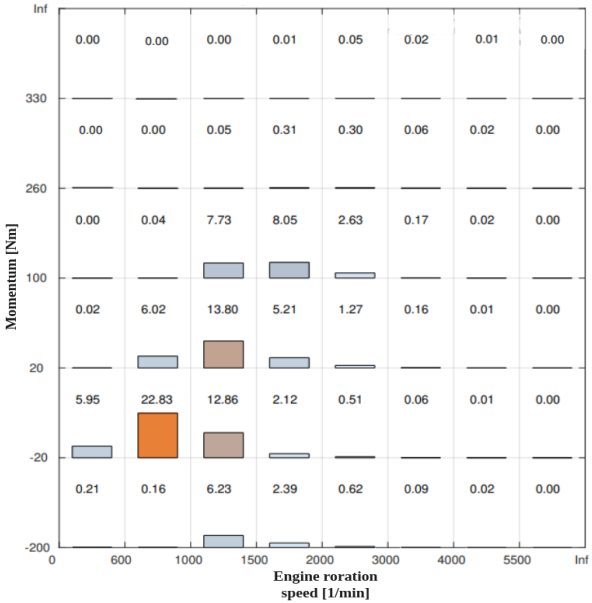
- A Total of 48 values per vehicle evaluation



Example: KF-PROZ multidimensional values

Validation of Multidimensional Values – KF-PROZ

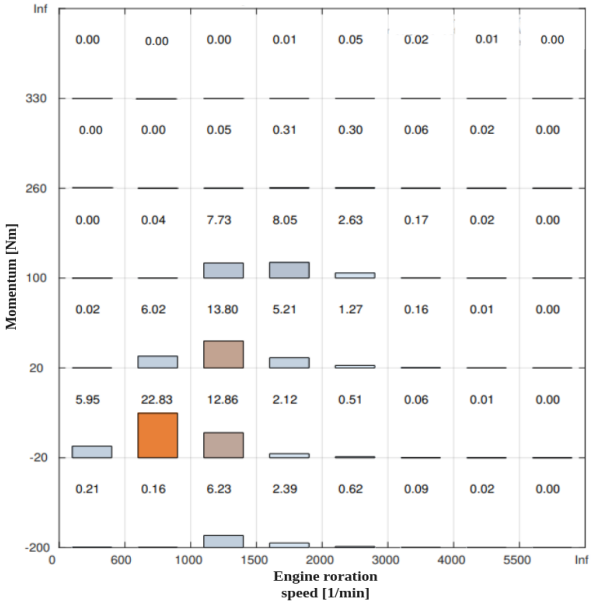
- A Total of 48 values per vehicle evaluation
- Engine variant specific



Example: KF-PROZ multidimensional values

Validation of Multidimensional Values – KF-PROZ

- A Total of 48 values per vehicle evaluation
- Engine variant specific
- Machine Learning classification approach

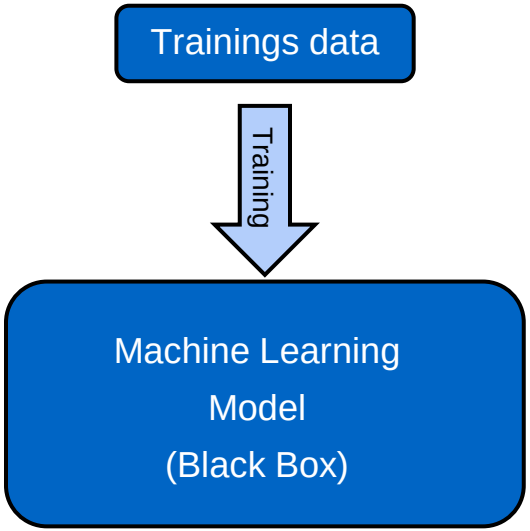


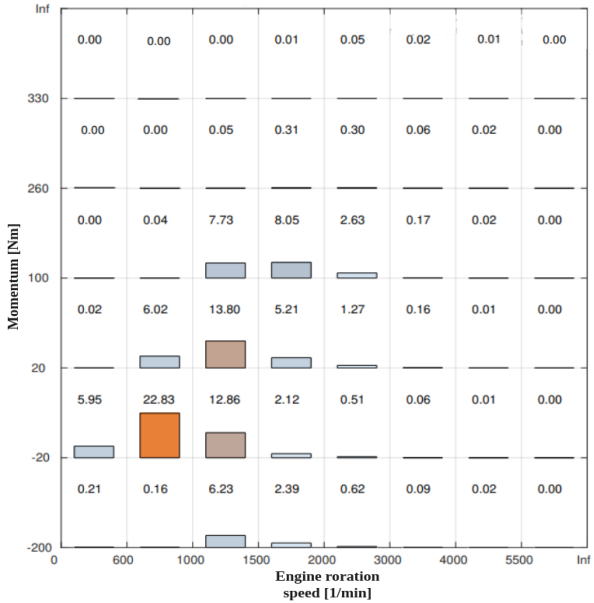
Example: KF-PROZ multidimensional values

Machine Learning Model (Black Box)

Validation of Multidimensional Values – KF-PROZ

- A Total of 48 values per vehicle evaluation
- Engine variant specific
- Machine Learning classification approach
- Trained with small portion of the FastaData (24,000 vehicle evaluations)

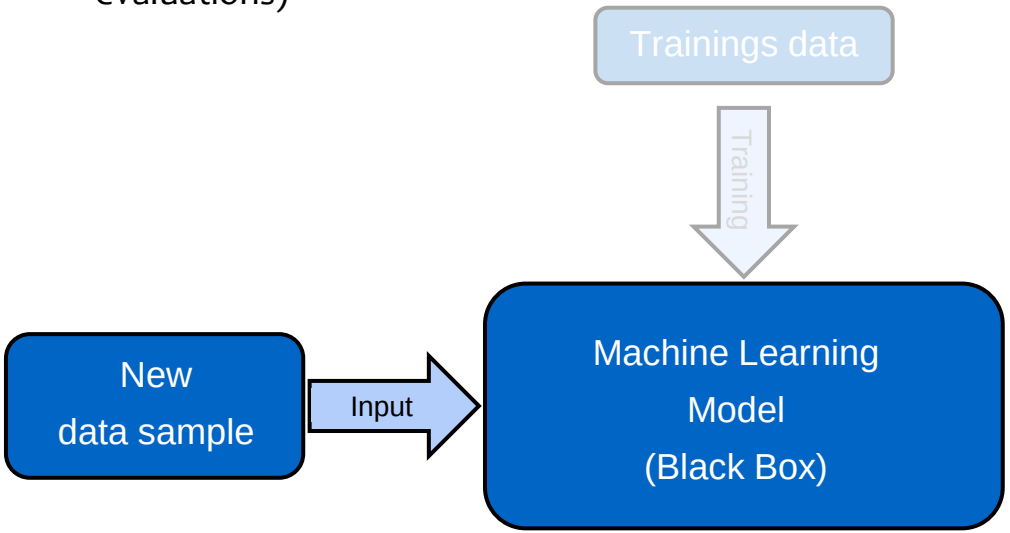


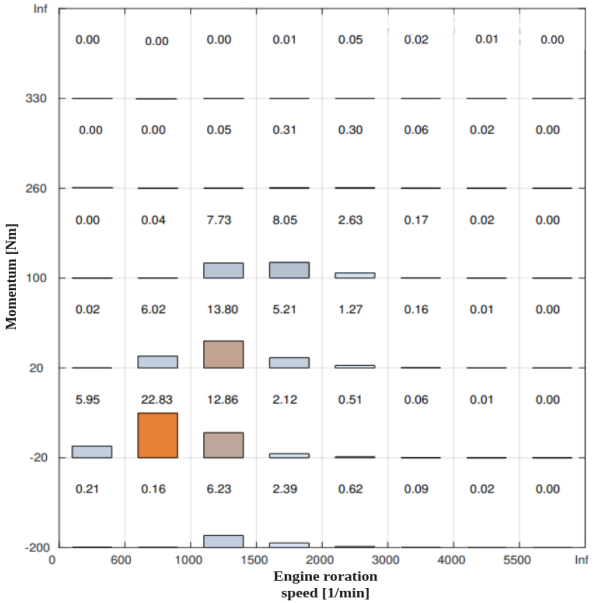


Example: KF-PROZ multidimensional values

Validation of Multidimensional Values – KF-PROZ

- A Total of 48 values per vehicle evaluation
- Engine variant specific
- Machine Learning classification approach
- Trained with small portion of the FastaData (24,000 vehicle evaluations)

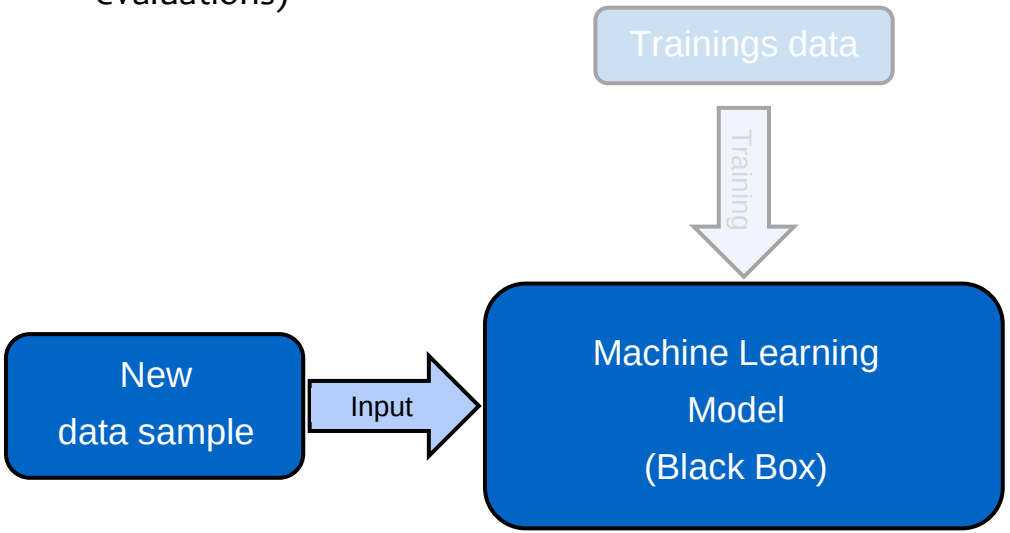


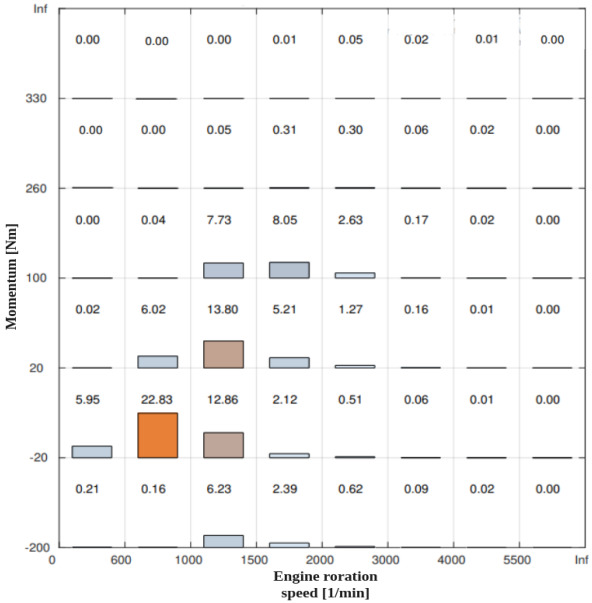


Example: KF-PROZ multidimensional values

Validation of Multidimensional Values – KF-PROZ

- A Total of 48 values per vehicle evaluation
- Engine variant specific
- Machine Learning classification approach
- Trained with small portion of the FastaData (24,000 vehicle evaluations)





Example: KF-PROZ multidimensional values

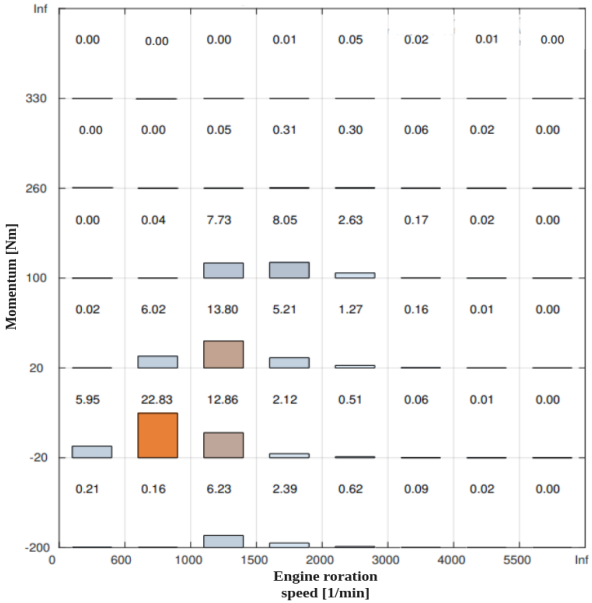
Motorvariante A

Motorvariante B

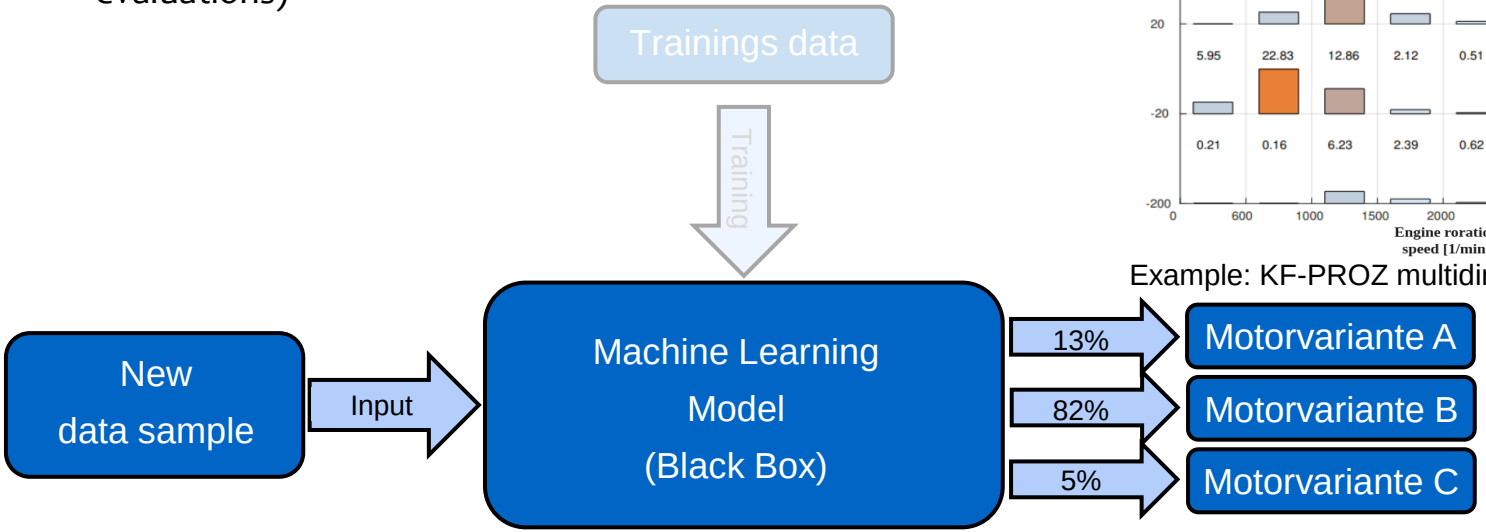
Motorvariante C

Validation of Multidimensional Values – KF-PROZ

- A Total of 48 values per vehicle evaluation
- Engine variant specific
- Machine Learning classification approach
- Trained with small portion of the FastaData (24,000 vehicle evaluations)

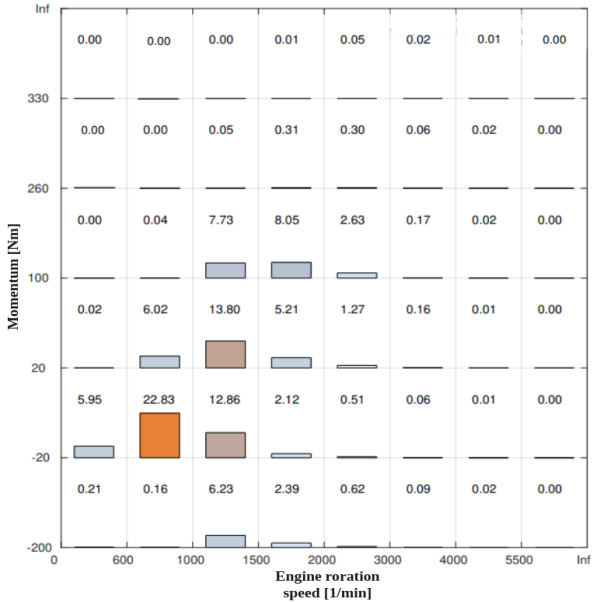


Example: KF-PROZ multidimensional values

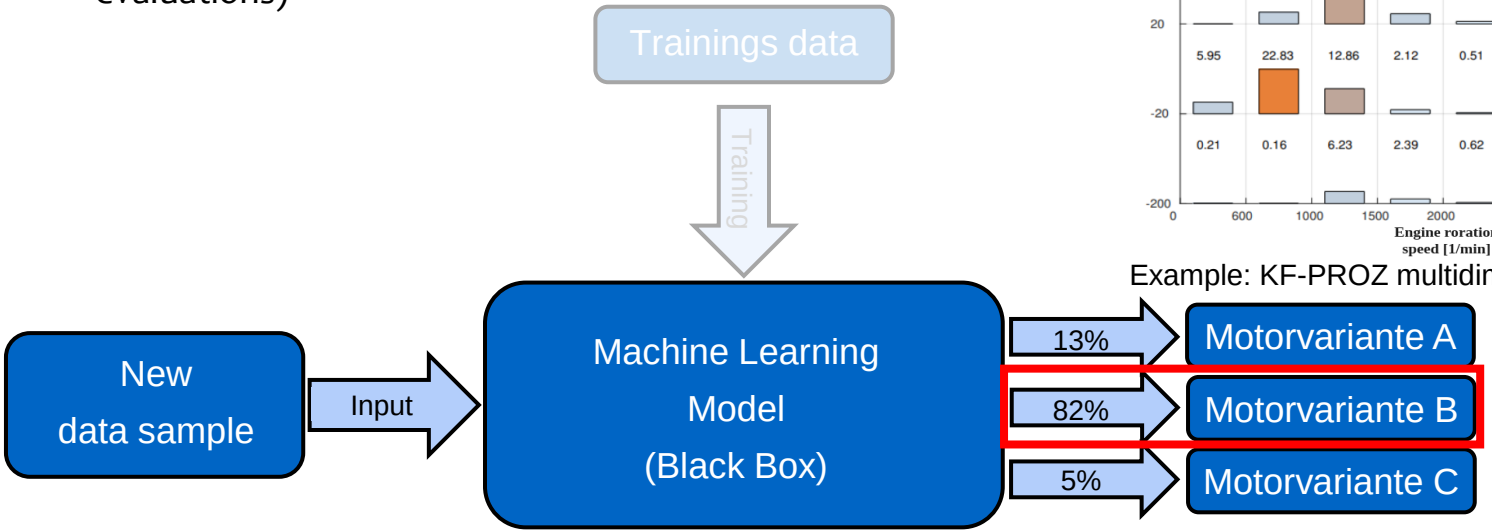


Validation of Multidimensional Values – KF-PROZ

- A Total of 48 values per vehicle evaluation
- Engine variant specific
- Machine Learning classification approach
- Trained with small portion of the FastaData (24,000 vehicle evaluations)

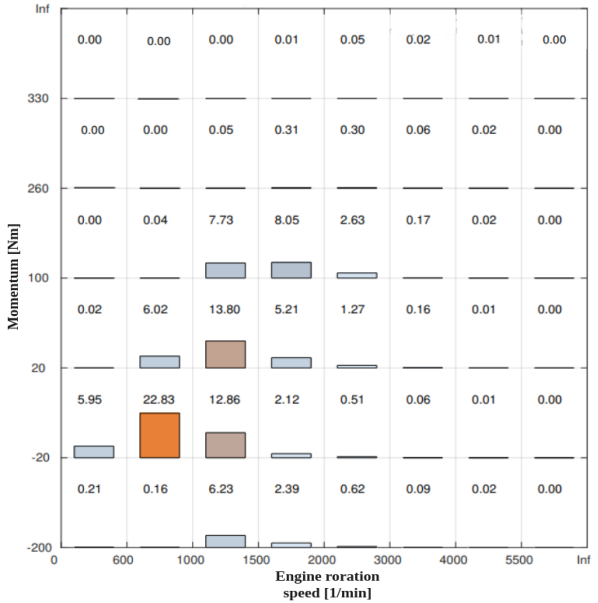


Example: KF-PROZ multidimensional values

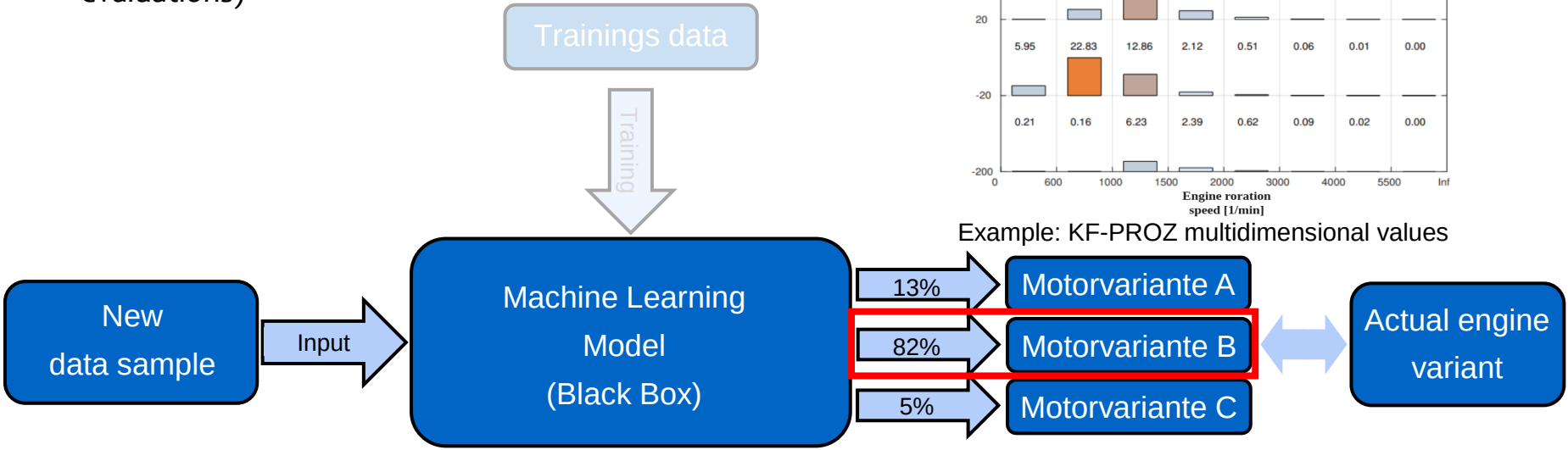


Validation of Multidimensional Values – KF-PROZ

- A Total of 48 values per vehicle evaluation
- Engine variant specific
- Machine Learning classification approach
- Trained with small portion of the FastaData (24,000 vehicle evaluations)



Example: KF-PROZ multidimensional values



Outline

- ~~Motivation~~
- ~~Problem Statement~~
- ~~Data Validation Requirements~~
- ~~Pandera (Data Validation Library)~~
- ~~Conditional Language~~
- ~~Validation of Multidimensional Values~~
- Evaluation
- Conclusion

Evaluation (1)

- **Setup:**
  - Intel Xeon Gold 6240 CPU (2.60 GHz, only 4 of the 16 cores used due to virtualization)
  - 132 GB DRAM
  - Taking the average of multiple benchmarks

Evaluation (1)

- **Setup:**
  - Intel Xeon Gold 6240 CPU (2.60 GHz, only 4 of the 16 cores used due to virtualization)
  - 132 GB DRAM
  - Taking the average of multiple benchmarks
- **ML Model - Multidimensional Values:**
  - ~ 97% accuracy

Evaluation (1)

- **Setup:**
  - Intel Xeon Gold 6240 CPU (2.60 GHz, only 4 of the 16 cores used due to virtualization)
  - 132 GB DRAM
  - Taking the average of multiple benchmarks
- **ML Model - Multidimensional Values:**
  - ~ 97% accuracy
- **Conditional Language** vs. **Direct implementation:**

Implementation	10,000 Records	50,000 Records	100,000 Records	1,000,000 Records
Direct	206.7 ms	975.6 ms	1954.6 ms	19179.3 ms
Conditional Language	225.3 ms	996.6 ms	1992.2 ms	19313.8 ms
Difference Δ of *Direct* and Conditional Language Implementation	18.6 ms	21.0 ms	37.6 ms	134.5 ms

Evaluation (2) – Pandera Validation

- **Pandera validation:**
  - In total 86 Validation rules
  - Combination of rules:
    - directly implemented and
    - formulated with the conditional language

Function	10,000 Records	50,000 Records	100,000 Records	1,000,000 Records
Schema validation	2.441 sec	9.296 sec	17.500 sec	173.362 sec

- Out of 10,000 random FASTA records 5.5% had at least one validation rule violation

Outline

- ~~Motivation~~
- ~~Problem Statement~~
- ~~Data Validation Requirements~~
- ~~Pandera (Data Validation Library)~~
- ~~Conditional Language~~
- ~~Validation of Multidimensional Values~~
- ~~Evaluation~~
- Conclusion

Conclusion

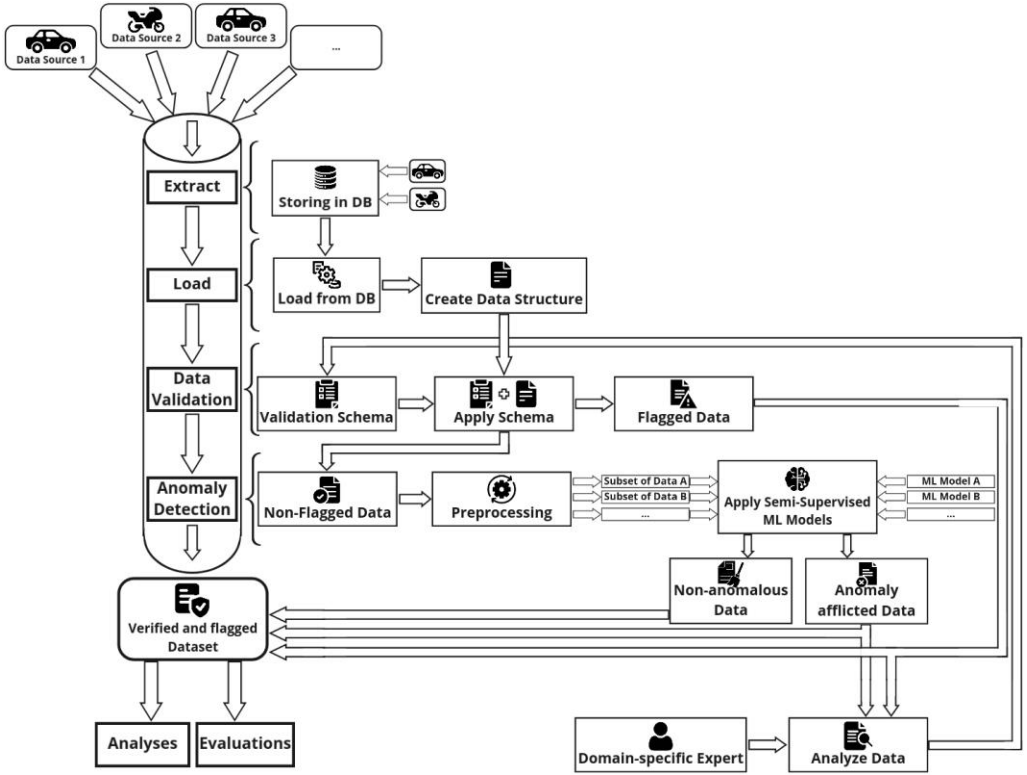
- FASTA data validation improves the overall data quality with a short runtime
- Multidimensional values validation rules are realizable by classification ML Models
- The created Conditional Language is universally applicable on logic tasks
- All processes can be ported to the Amazon Web Services (AWS) Cloud

Sources

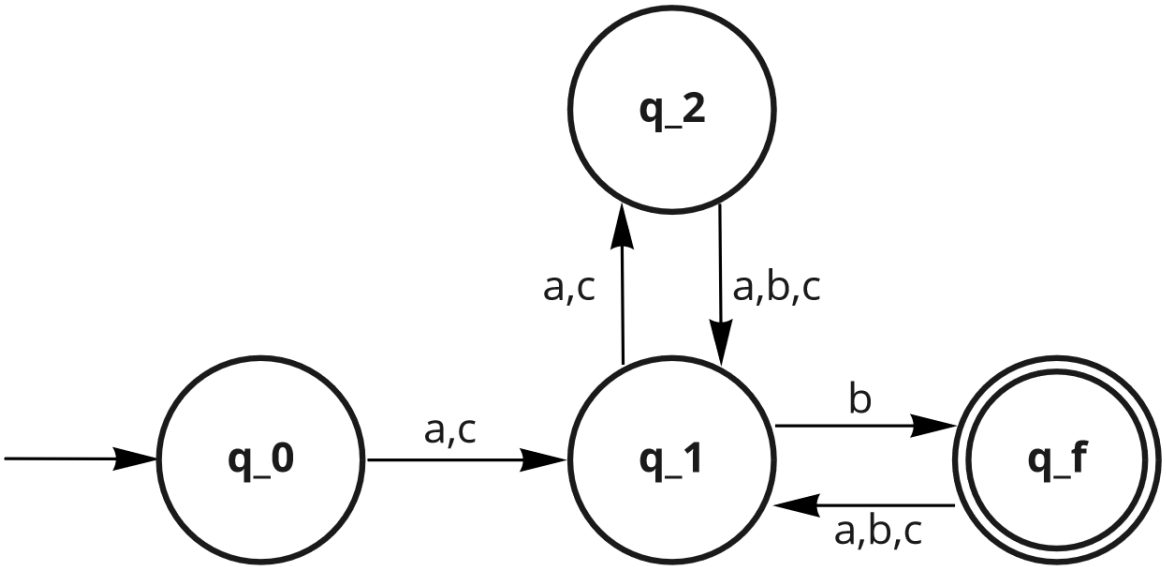
- N. Bantilan, „Pandera: Statistical Data Validation of Pandas Dataframes“, in Proceedings of the 19th Python in Science Conference, M. Agarwal and C. Calloway and D. Niederhut and D. Shupe, 2020, pp. 116 – 124
- W. McKinney, „Data Structures for Statistical Computing in Python“, in Proceedings of the 9th Python in Science Conference, S. van der Walt and J. Millman, 2010, pp. 56 – 61
- A. Gelron, „Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to build Intelligent Systems“, 2nd Edition, O‘Reilly, 2019, September 2019
- M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M, Wicke, Y. Yu and X. Zheng. „TensorFlow: Large-scale machine learning on heterogeneous systems“, 2015. Software available from tensorflow.org. Last accessed 10.09.2022.
- Icons from Flaticon.com. Artists: Becris, Freepik, Kiranshastry, Payungkead, Pause08, ultimatearm

Backup

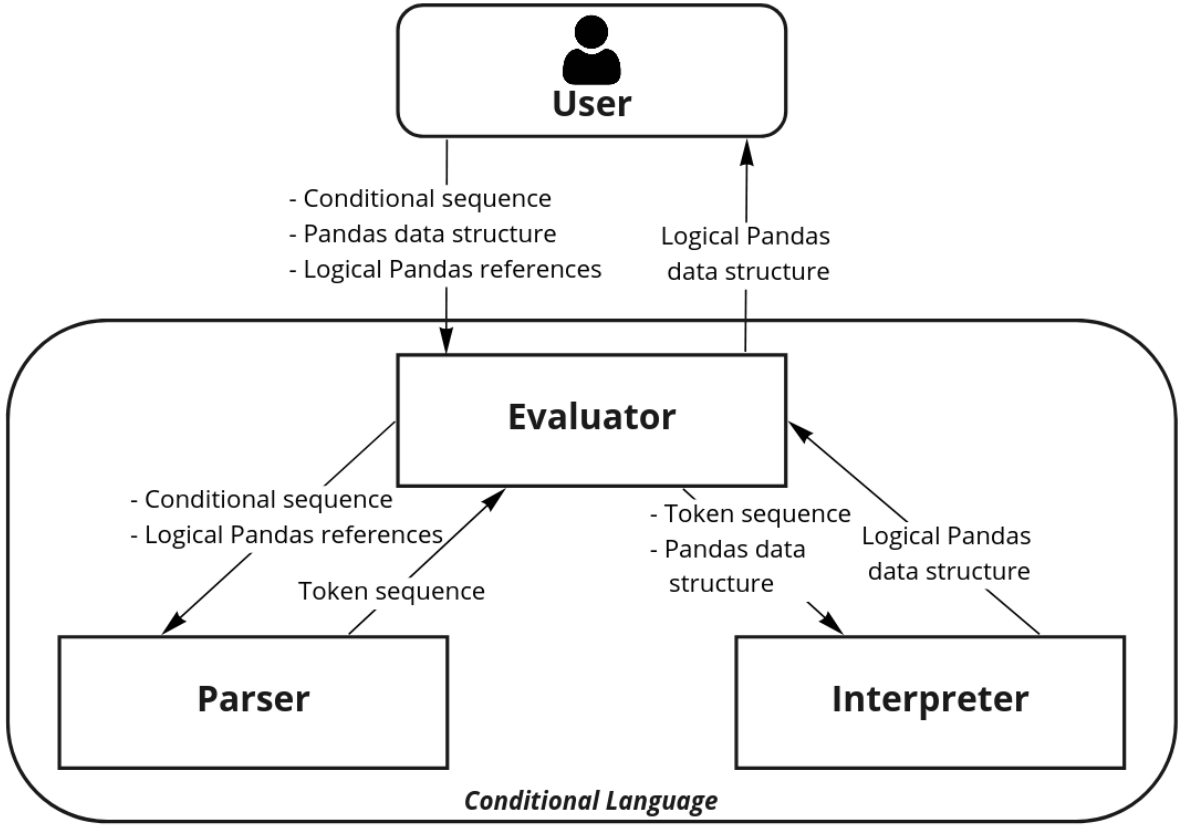
Backup – Overview



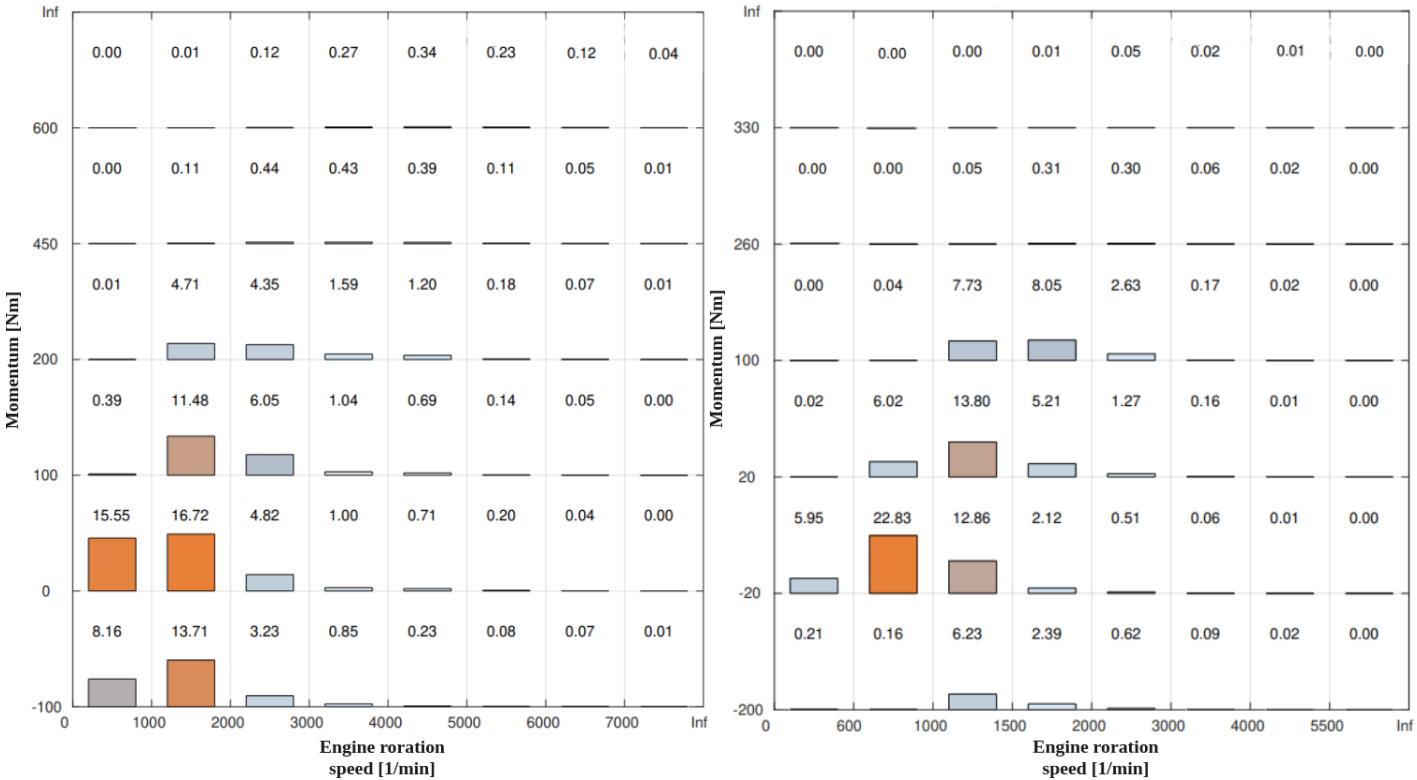
Backup – DSL Automata explained



Backup – DSL Explanation of the Evaluator Abstraction



Backup – Multidimensional Values KF-Proz Comparison



Backup – Evaluation(s)

- Multidimensional (trained) classification ML Model:
  - (Input) – 200 – ReLu – 200 – ReLu – (Output)

- Found violations of validation rules of 10,000 random FASTA data records:
  - ~ 85 % NULLs
  - ~ 5 % not summed to 100
  - ~ 3 % Incorrect engine variant distribution (KF-Proz)
  - ~ 2 % each :
    - Presented OperatingDays conditional validation rule
    - Date before current year
    - Not in specified range
    - Pattern match violation
  - ~ 1 % smaller than 0