DAIKIRI
Diagnostische KI für industrielle Daten

DAIKIRI
Erklärbare Diagnostische KI für industrielle Daten

Project Number: 01IS19085 **Start Date of Project:** 01/01/2020 **Duration:** 30 months

Deliverable 6.2: Prototypical Implementation and Evaluation

Dissemination Level	Public
Due Date of Deliverable	Month 30, 30/06/2022
Actual Submission Date	Month 30, 30/06/2022
Work Package	WP6 — Use Cases
Task	T6.12
Type	Report
Approval Status	Final
Version	1.0
Number of Pages	17

Abstract: This deliverable presents the results of the final and successful prototypical implementation of the DAIKIRI pipeline for a dedicated use case in the field of logistics. Further, we present another use case and its data analysis. However, for this case it turned out, that the DAIKIRI pipeline has no value due to the defined hypotheses and the data characteristics.

The information in this document reflects only the author's views and the Federal Ministry of Education and Research (BMBF) is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

This project has received funding from the Federal Ministry of Education and Research (BMBF) within the project DAIKIRI under the grant no 01IS19085.

History

Version	Date	Reason	Revised by
0.0	14/06/2022	First draft created	Carolin Walter
0.1	20/06/2022	Draft revised	Jean Chorin
0.2	28/06/2022	Draft revised	Mohammad Sajjadi
0.3	30/06/2022	Final version created	Martin Voigt
1.0	30/06/2022	Final version submitted	Carolin Walter

Author List

Organization	Name	Contact Information
elevait	Jean Chorin	jean.chorin@elevait.de
elevait	Mohammad Sajjadi	mohammad.sajjadi@elevait.de
elevait	Martin Voigt	martin.voigt@elevait.de
USU	Carolin Walter	carolin.walter@usu.com

Contents

1 Introduction

2 Using the DAIKIRI Platform for Smart Logistics Use Case

Up to now the components of the DAIKIRI pipeline were developed independently of the use case and evaluated mainly on benchmarking datasets. In this section we describe what has to be done to adapt the pipeline of the DAIKIRI platform to a specific use case, in our example the smart logistics use case. The smart logistic use case is described in Deliverable 6.1. We use the complete pipeline where the white box approach is used in combination with the black box approach and the SML approach. This pipeline is depicted in Figure 1.

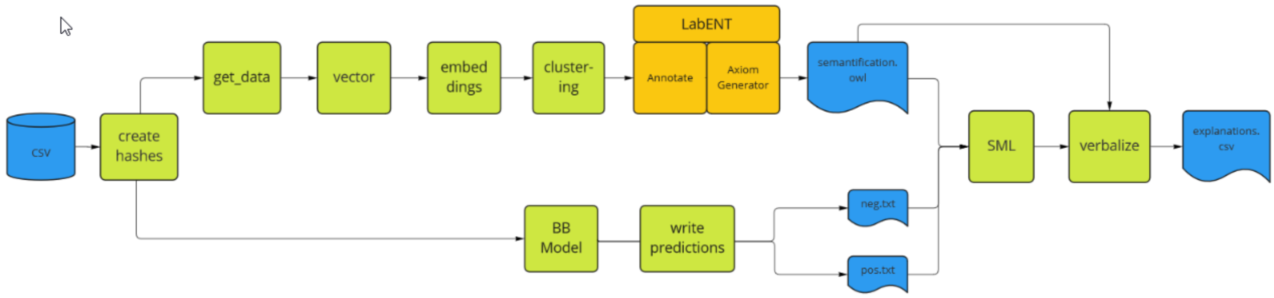


Figure 1: The figure shows a DAIKIRI pipeline with the combination of black-box and white-box approach and SML.

In the course of implementing the smart logistics use case in the DAIKIRI pipeline, we found that a non-variable index for the data points is helpful. Otherwise, the naming of the events does not remain consistent if they are numbered chronologically as soon as the data record under consideration changes, for example when splitting the data record into train, test and validation set. Therefore, a hash is now created directly for each data point in the CSV file. This is an MD5 hash that is calculated across all features of the data point. In the ontology, this hash is then the name of the individual.

2.1 Create an Ontology from CSV

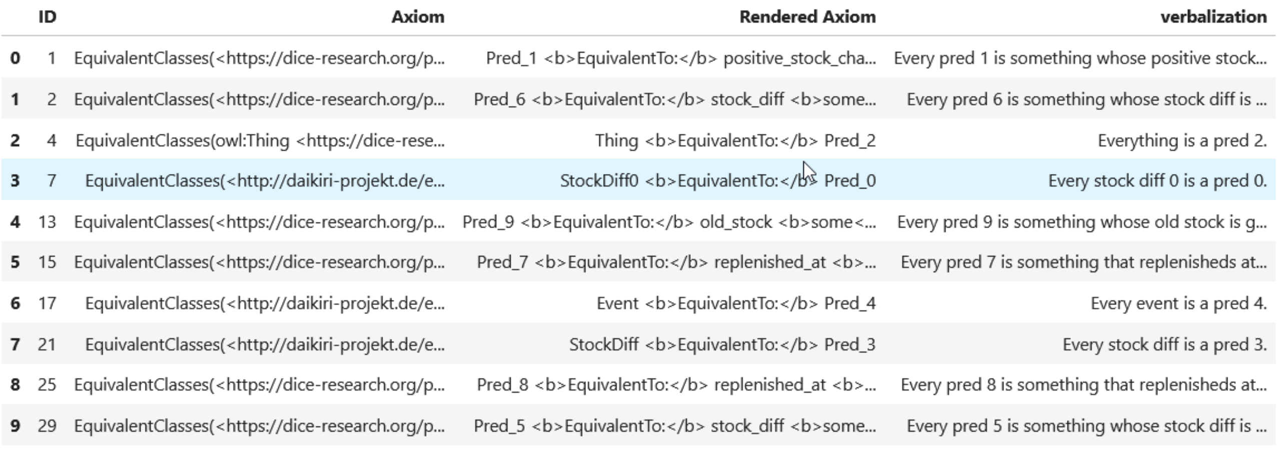
As part of the white box approach (upper part in figure 1), an ontology is created for the data from the CSV file. In the combined pipeline, the data of the test data set are used in the white box approach, as the predictions are made on this set in the black box approach to determine the positive and negative examples. Thereby the pipeline is used in an unsupervised mode. The names of the individuals in the ontology must include the individuals of the positive and negative examples. The dataframe of the test dataset is vectorized using the Vectograph library. The embeddings are then calculated. For this purpose, we used the Knowledge Graph Embedding Model SHALLOM for the Smart Logistic Use Case. The clusters on the embeddings of the Logistic Use Case were calculated via kmeans. With LabENT, the clusters can then be labeled. And from this the ontology is created. Since the structure of the data is very flat, further cross-connections between the entities, the object properties, and the literals, the data properties, were calculated at this point and inserted into the ontology. This ontology is then used as one of the inputs for the SML and verbalization. It is stored under the name semantification.owl.

2.2 Label the data points

Parallel to the white box pipeline, the black box approach (lower part in figure 1) can be run. In this way the input data set does not need to be labeled. For this purpose, a suitable black box model must first be found, which can be trained on the existing data set and whose predictions have a good accuracy. This black box model for the Smart logistic use case was developed and is described in Deliverable 4.3. As part of the pipeline, the model is trained on the training data set. The learned model is then used to make the predictions for the test datas et. The hashes for data points with predicted anomalies are stored in a file named pos.txt. The hashes of the normal cases are written to the file neg.txt.

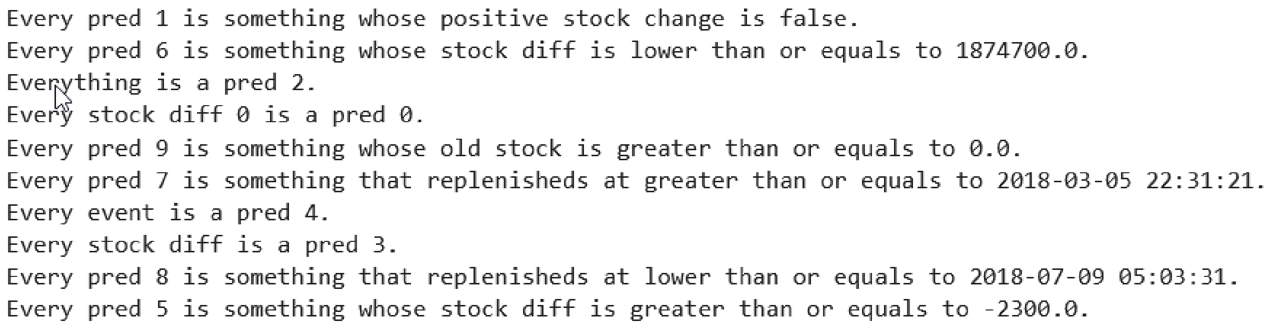
2.3 Get the explanations from SML

Now the three inputs for the SML approach (rear part in Figure 1) are available. With the help of the Onotlearn library, an intrinsically explainable model is learned from ontology and the positive and negative examples. The outputs of the model are converted into natural language via the RAKI Verbalizer. This explains the SML model. The explanations for the model of the Logistic Use Case are shown in Figure 2 and Figure 3. Each line shows a different concept. These explanations are stored in a CSV file after the combined pipeline has been completed.



	ID	Axiom	Rendered Axiom	verbalization
0	1	EquivalentClasses(<https://dice-research.org/p...	Pred_1 <b>EquivalentTo:</b> positive_stock_cha...	Every pred 1 is something whose positive stock...
1	2	EquivalentClasses(<https://dice-research.org/p...	Pred_6 <b>EquivalentTo:</b> stock_diff <b>some...	Every pred 6 is something whose stock diff is ...
2	4	EquivalentClasses(owl:Thing <https://dice-rese...	Thing <b>EquivalentTo:</b> Pred_2	Everything is a pred 2.
3	7	EquivalentClasses(<http://daikiri-projekt.de/e...	StockDiff0 <b>EquivalentTo:</b> Pred_0	Every stock diff 0 is a pred 0.
4	13	EquivalentClasses(<https://dice-research.org/p...	Pred_9 <b>EquivalentTo:</b> old_stock <b>some<...	Every pred 9 is something whose old stock is g...
5	15	EquivalentClasses(<https://dice-research.org/p...	Pred_7 <b>EquivalentTo:</b> replenished_at <b>...	Every pred 7 is something that replenisheds at...
6	17	EquivalentClasses(<http://daikiri-projekt.de/e...	Event <b>EquivalentTo:</b> Pred_4	Every event is a pred 4.
7	21	EquivalentClasses(<http://daikiri-projekt.de/e...	StockDiff <b>EquivalentTo:</b> Pred_3	Every stock diff is a pred 3.
8	25	EquivalentClasses(<https://dice-research.org/p...	Pred_8 <b>EquivalentTo:</b> replenished_at <b>...	Every pred 8 is something that replenisheds at...
9	29	EquivalentClasses(<https://dice-research.org/p...	Pred_5 <b>EquivalentTo:</b> stock_diff <b>some...	Every pred 5 is something whose stock diff is ...

Figure 2: The figure shows the output of the verbalizer.



```
Every pred 1 is something whose positive stock change is false.
Every pred 6 is something whose stock diff is lower than or equals to 1874700.0.
Everything is a pred 2.
Every stock diff 0 is a pred 0.
Every pred 9 is something whose old stock is greater than or equals to 0.0.
Every pred 7 is something that replenisheds at greater than or equals to 2018-03-05 22:31:21.
Every event is a pred 4.
Every stock diff is a pred 3.
Every pred 8 is something that replenisheds at lower than or equals to 2018-07-09 05:03:31.
Every pred 5 is something whose stock diff is greater than or equals to -2300.0.
```

Figure 3: The figure shows how the explanations look like.

2.4 Other modes of usage

The methods of the black box and white box approach can also be used separately from each other. Then the white box pipeline requires some labeled instances on which the SML can learn. These labels can be included in the ontology. The standalone version of the white box pipeline is depicted in 4.

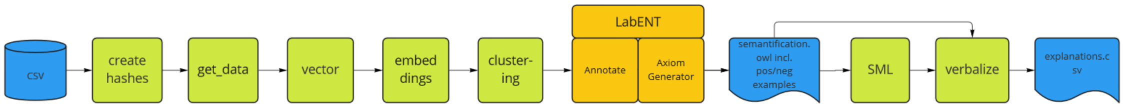


Figure 4: The figure shows a supervised mode of the DAIKIRI pipeline for the white box approach.

The black box pipeline can also be used for itself. For this purpose, we have tried out existing methods for the explainability of black box models shown in Deliverable D4.3. In the independent black box pipeline, we have implemented SHAP as an explainer. The resulting pipeline is shown in 5.

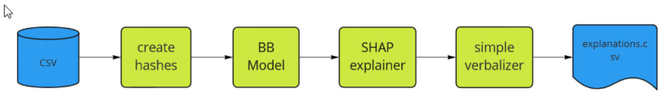
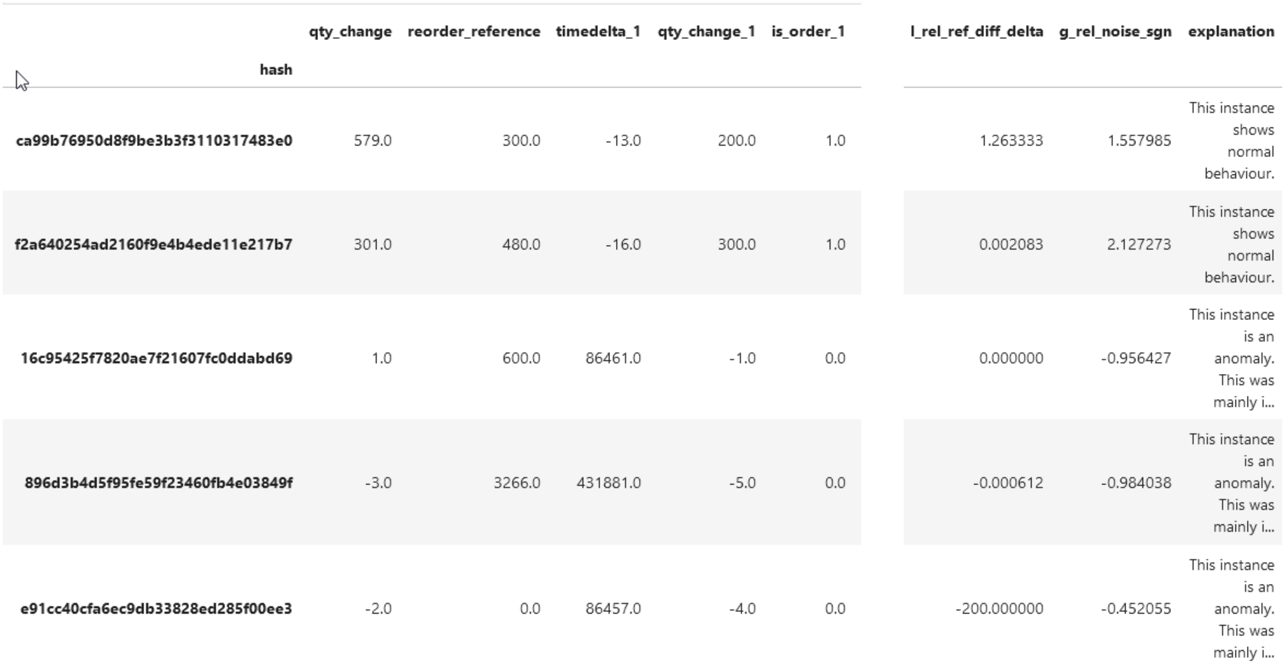


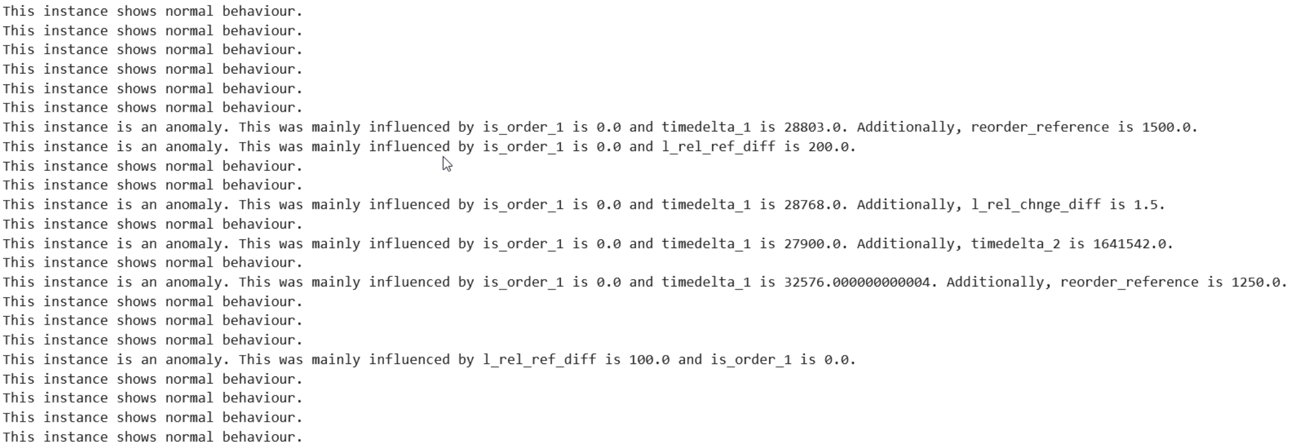
Figure 5: The figure shows the DAIKIRI pipeline in a stand alone version of the black box approach.

Individual data points of the test data set are explained here. Since SHAP does not provide adequately readable explanations for non-ML-specialist, we have further processed the SHAP values in a post-processing step. It displays only the most important features that are critical to a prediction and the value of the feature at that data point. The output is converted rule-based to natural language and stored in a CSV file. Some of the output is shown in Figure 6 and Figure 7.



hash	qty_change	reorder_reference	timedelta_1	qty_change_1	is_order_1	l_rel_ref_diff_delta	g_rel_noise_sgn	explanation
ca99b76950d8f9be3b3f3110317483e0	579.0	300.0	-13.0	200.0	1.0	1.263333	1.557985	This instance shows normal behaviour.
f2a640254ad2160f9e4b4ede11e217b7	301.0	480.0	-16.0	300.0	1.0	0.002083	2.127273	This instance shows normal behaviour.
16c95425f7820ae7f21607fc0ddabd69	1.0	600.0	86461.0	-1.0	0.0	0.000000	-0.956427	This instance is an anomaly. This was mainly i...
896d3b4d5f95fe59f23460fb4e03849f	-3.0	3266.0	431881.0	-5.0	0.0	-0.000612	-0.984038	This instance is an anomaly. This was mainly i...
e91cc40cfa6ec9db33828ed285f00ee3	-2.0	0.0	86457.0	-4.0	0.0	-200.000000	-0.452055	This instance is an anomaly. This was mainly i...

Figure 6: The figure shows the output of the verbalizer.



```
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance is an anomaly. This was mainly influenced by is_order_1 is 0.0 and timedelta_1 is 28803.0. Additionally, reorder_reference is 1500.0.
This instance is an anomaly. This was mainly influenced by is_order_1 is 0.0 and l_rel_ref_diff is 200.0.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance is an anomaly. This was mainly influenced by is_order_1 is 0.0 and timedelta_1 is 28768.0. Additionally, l_rel_chnge_diff is 1.5.
This instance shows normal behaviour.
This instance is an anomaly. This was mainly influenced by is_order_1 is 0.0 and timedelta_1 is 27900.0. Additionally, timedelta_2 is 1641542.0.
This instance shows normal behaviour.
This instance is an anomaly. This was mainly influenced by is_order_1 is 0.0 and timedelta_1 is 32576.000000000004. Additionally, reorder_reference is 1250.0.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance is an anomaly. This was mainly influenced by l_rel_ref_diff is 100.0 and is_order_1 is 0.0.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance shows normal behaviour.
This instance shows normal behaviour.
```

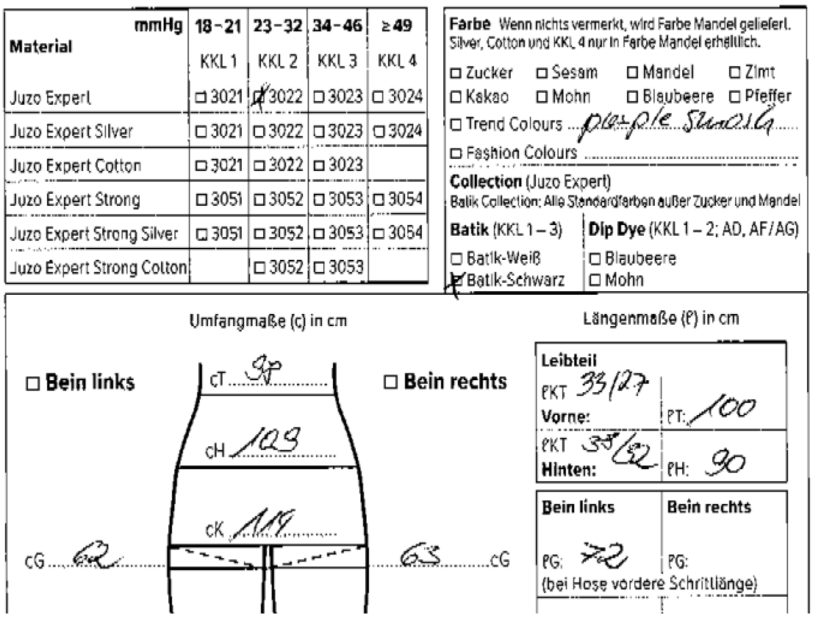
Figure 7: The figure shows how the explanations look like.

3 Analysis and Classification of Documents

The 2nd use case provided by elevait is based on the product **in:forms**. Here, information is extracted from template-based documents. In our productive deployments with customers we see several quality issues in the documents (bad handwriting's or scans) or how the documents are filled, so that validation rules are hit. The idea of the use case is to identify automatically clusters of documents which have dedicated quality issues in common. This migh give the possibility to address these in detail and over-time to improve the automation rate for the machine learning based information extraction.

3.1 Description of the data

elevait in:forms allows customers to send scanned forms as image or PDF to extract information automatically. Such information include manually written text, if a checkbox was checked or not, or just to classify content on other dimensions if needed. Each piece of information is given a semantic class from a predefined set of ontologies, for instance the one called *numeric:ck* (a numeric value extracted from text) or *boolean:farbe* (the state of a checkbox). Because the extractions are performed by machine learning models, a prediction confidence is also computed for each field. Finally, inference and/or validation shapes can be applied on the extracted values. The customers can leverage multiple forms, with different text fields and/or checkboxes. Thus, an incoming form must first be matched to an available template. A cropped example of such form can be found in Figure 8.



Material mmHg	18–21 KKL 1	23–32 KKL 2	34–46 KKL 3	≥49 KKL 4
Juzo Expert	□ 3021	☒ 3022	□ 3023	□ 3024
Juzo Expert Silver	□ 3021	□ 3022	□ 3023	□ 3024
Juzo Expert Cotton	□ 3021	□ 3022	□ 3023	
Juzo Expert Strong	□ 3051	□ 3052	□ 3053	□ 3054
Juzo Expert Strong Silver	□ 3051	□ 3052	□ 3053	□ 3054
Juzo Expert Strong Cotton		□ 3052	□ 3053	

Farbe Wenn nichts vermerkt, wird Farbe Mandel geliefert.
Silver, Cotton und KKL 4 nur in Farbe Mandel erhältlich.

□ Zucker □ Sesam □ Mandel □ Zimt
□ Kakao □ Mohn □ Blaubeere □ Pfeffer
□ Trend Colours purple Sunday
□ Fashion Colours ....

Collection (Juzo Expert)
Batik Collection: Alle Standardfarben außer Zucker und Mandel

Batik (KKL 1 – 3)
□ Batik-Weiß
☑ Batik-Schwarz

Dip Dye (KKL 1 – 2; AD, AF/AG)
□ Blaubeere
□ Mohn

Umfangmaße (c) in cm

□ **Bein links**
□ **Bein rechts**

cT 98
cH 109
cK 119
cG 62
63 cG

Längenmaße (ℓ) in cm

Leibteil
ℓKT 33/27
Vorne:
ℓT: 100
ℓKT 35/32
Hinten:
ℓH: 90

Bein links	**Bein rechts**
ℓG: 72	ℓG:

(bei Hose vordere Schrittlänge)

Figure 8: Example of a form with text fields as boxes and checkboxes.

The actual process of data extraction is as follow:

1. A scanned form is sent as image or PDF;
2. The template which matches against the form layout is selected;
3. The text in the text boxes specified by the templates are detected and recognized (OCR), as well as the status of the checkboxes (checked or not);

Id	dateCreated	templateId	customerID	*class*	inference/*UUID*	validation/*UUID*	...
6ed9	2021-08-19	4ad2	12121212	0.91	0	NaN	...
f6e3	2021-08-05	f6e3	12345678	0.63	NaN	1	...
d622	2021-08-26	d1c9	98989898	0.54	NaN	NaN	...

Table 1: Simplified example of the content of the data

4. The inference shapes are applied. The shape can either succeed or fail, because some input was ill-formed. A value can be computed if the inputs were suitable;

5. The validation shapes are applied. The shape can either succeed or fail, because some input was ill-formed;

6. All the aforementioned values are stored in a database, along with the original form.

In the end, for each document various information is available, which is predefined by the semantic annotation of the template. For this use case, we are not interested in the content of each data point but more on the quality of extraction. This is related to the confidence of the applied machine learning models. For the use case, we reduce the dataset to the following attributes.

Id the UUID of the document

dateCreated the timestamp of the extraction

templateId the ID of the template which matches the form

customerNumber the ID of the customer who has sent the form

semantic classes for each semantic class, the confidence for the extraction (between 0 and 1), or NaN if a semantic class is not linked to the template of the document

inference-shape/*UUID* for each shape, a value of NaN, 0 or 1 is stored. NaN means the shape is not applied on the current template, 0 means the inference could be performed, 1 means an error occurred

validation-shape/*UUID* for each shape, a value of NaN, 0 or 1 is stored. NaN means the shape is not applied on the current template, 0 means the validation was successful, 1 means it was not.

The complete information was exported to a CSV file which is the foundation of the analyses described below. A mock example of the content of the CSV file is shown in Table 1. Just note that a single semantic class, inference shape and validation shape are shown and that the date timestamps are simplified, as well as the UUIDs.

The actual final CSV file contains 62064 rows (one per document) and 3273 columns. There are 158 semantic classes, 576 validation shapes, and 2535 inference shapes. Due to the amount of inference shapes being quite high compared to the validation shapes and semantic classes, they will be removed from all operations to simplify the computations and analyses.

3.2 First hypothesis

The first hypothesis was that some customers systematically input wrong data, which creates extraction errors. This would, in turn, make the validation shapes being violated and the inference shapes not computed.

Thus, the original goal was to cluster the customers between the ones who often generate errors or violations with the shapes, the ones who do it with a smaller probability, and the ones who do it with a very low probability.

3.3 First analysis

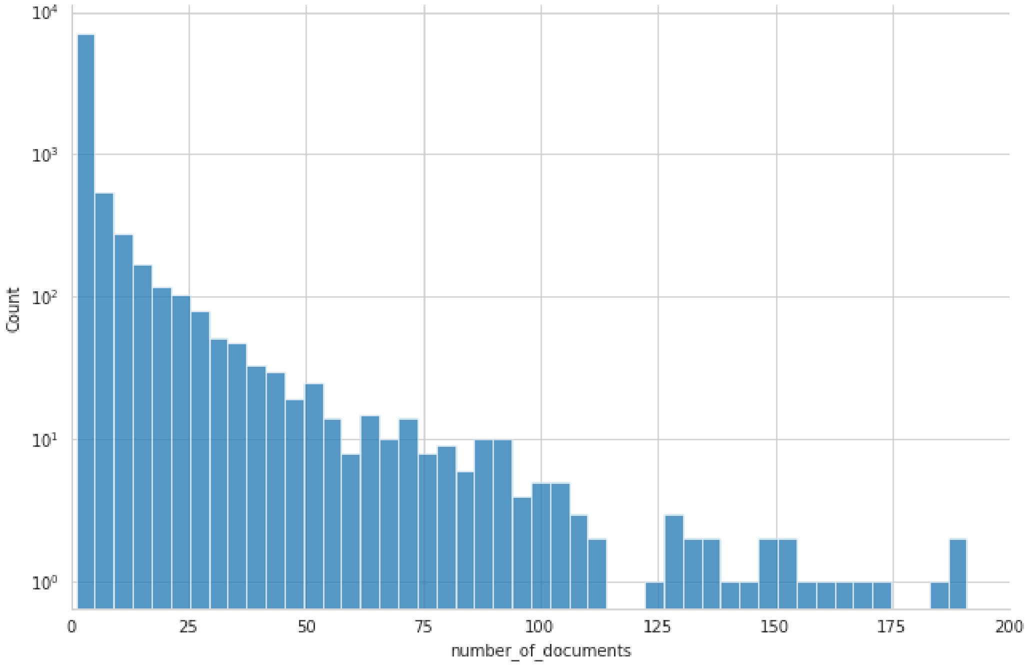


Figure 9: Distribution of the number of documents per customer.

For a start, the amount of documents that each customer sent was computed. With these values, the amount of customer depending on the number of document can be plotted, see Figure 9 (with a logarithmic scale). We can see that most of the customers actually sent very few documents, roughly between 1 and 5. Also, few customers sent more than 55 documents. For most of the amount of documents sent above 55, only around 10 customers or less belong to each category.

For each validation shape applied on all documents, the distribution of the shapes with and without violation can be seen on Figure 10. Basically, for most shapes, no violation was found. Only a small amount of them were violated across all documents, around 5%.

However, if we group all documents which had at least one shape violated together, and all documents without any, the distribution obtained is presented on Figure 11. Most documents (around 30.000) had no rule violated, but close to 20.000 had at least one. Thus, almost 40% of all documents contain at least one error in the values extracted from the semantic classes, which led to a shape being violated.

Due to the hypothesis, our goal can be roughly defined as finding a correlation between each of

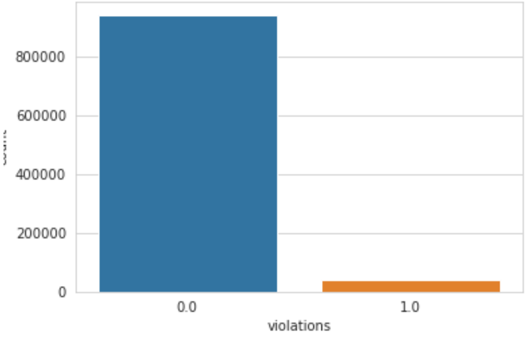


Figure 10: Distribution of the shapes for all documents without (0) and with violations (1).

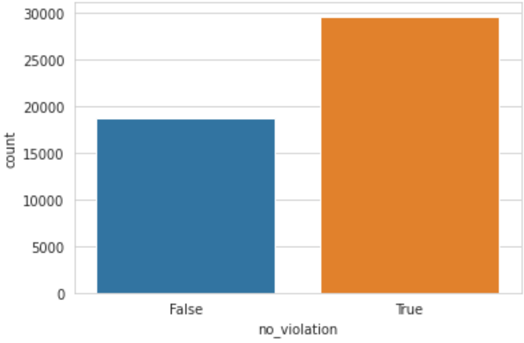


Figure 11: Distribution of documents without any violations (True), and with violations (False).

the customers and the amount of shapes violated in the documents they provided. Thus, we should find if there is an influence of the customer on distribution of the violated shapes. If so, we are also interested in how the distribution is affected. To verify if we have a dependency, we created a second dataset, based on a copy of the original one. The only difference in this new one is that the column of *customerId* is randomized. Every document is affected to a random customer, but the distribution of customer is kept. This means that the amount of document for each customer stays the same, only the information of which document it sent is changed. In the remainder of the chapter, the original dataset will be defined as having "the original distribution", and the new dataset as having "the randomized distribution".

As a result, the distribution of the violated shapes changed as well for each customer in the randomized distribution. For each customer, for both datasets, the average amounts of violations across all documents of a customer were computed. The resulting histogram of the counts of these averages can be found on Figure 12 for the original distribution and Figure 13 for the randomized distribution. We can observe some difference: in the second case, the averages are closer to 0, while they are a bit more spread out in the first case. Indeed, most customers have less than 3 violations on figure 13, while the figure 15 shows that a large part of customers can have up to 5 violations on average. This tends to imply that some customers generated in general a higher amount of violated shapes, compared to the same customers distributed randomly. As a side note, the high bars which form a regular pattern are most probably artifacts obtained as a results of doing averages and other statistical operations.

Before further observing the differences between the datasets, we focused on the distribution of the customers depending on the ratio of documents with at least one violated shape. This is displayed

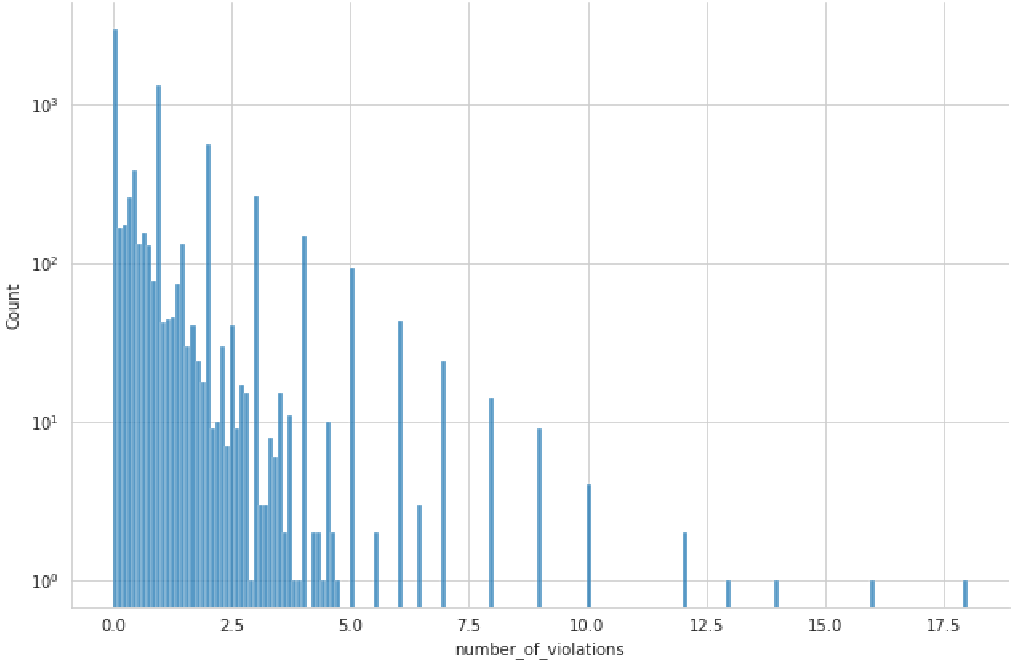


Figure 12: Distribution of the number of customer for each average number of violations (with original distribution).

on Figure 14 for the original distribution. On the left, the first column shows the amount of customers with documents with at least one violated shape. On the right, the last column shows the amount of customers with documents without any violated shape. In the middle, the columns around the average of 0.5 represents the customers who have half of their documents with at least one shape violated. As easily seen, most of the customers have either all documents with violated shapes, or none. After further analysis, this situation derived from the facts that most customers sent very few documents (see Figure 9). Thus, the threshold of 25 documents was chosen: we consider in the rest of the first analysis only customers who sent at least 25 documents. Below this number, we cannot have any good statistical behavior because the sample size is too small.

With this new threshold, we computed the distributions of the customers depending on violation shapes ratio and number of documents. The results are presented on Figure 15 for the original distribution and Figure 16 for the randomized distribution. The color of each hexagon represents the amount of customers presents on the surface of the hexagon. On the surface drawn by the hexagons with a darker shade, the density of documents is higher than on hexagons with a brighter shade.

Regarding the original distribution, we can observe that most customers have sent less than 50 documents. Also, a large part of the documents sent have a ratio of documents without any shape violated between 60% and 80%. The randomized distribution, on the other hand, have most documents with a ratio between 50% and 70%. Also, the distribution in general seems more spread out, with less central hexagons gathering all documents. The amount of less than 50 documents sent per customer is kept, which was expected, as only the attribution of customer to document was randomized, not the amount of document per customer.

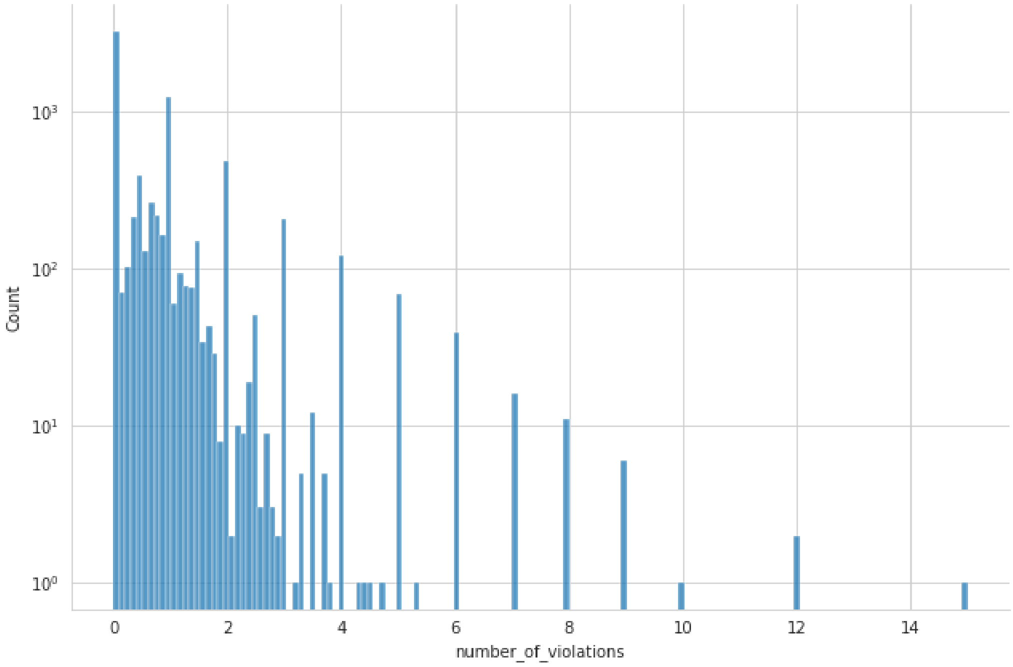


Figure 13: Distribution of the number of customer for each average number of violations (with randomized distribution).

3.4 Interpretation of the results of the first analysis

By comparing Figures 15 and 16, we concluded that if we take a document given by a randomly attributed customer, its probability to have a ratio of documents with at least one violation shape is lower than with the originally attributed customer. This means that the customers have an influence on the violation shapes: for some customers, there is in general a higher probability that the document they send have at least one validation shape violated. Thus, some customers seem to send documents with errors with a higher probability.

However, also by comparing Figures 15 and 16, we can observe that the differences between the two distributions are not large enough: there is no clear separation. The trend of customers sending more failing documents is somehow diluted in the averages of violated shapes and depends also on the number of documents sent (cf the chosen threshold of 25).

3.5 Second hypothesis

Therefore, the second part will be focused on finding a relationship between the templates and the confidences of the semantic classes.

Each template has a defined list of semantic classes it contains. The classes can be present on multiple templates. For each of these classes, a confidence on the extraction process can be computed for a specific document. Thus, we can compare templates to find the ones where customers often made mistakes. These mistakes then can lead to validation shapes failing.

Hence, we make the hypothesis that some classes are linked to fields which can be difficult to extract, because the fields are too small, or the templates are misleading for the person filling them. We search for classes with low confidence over a large amount of documents.

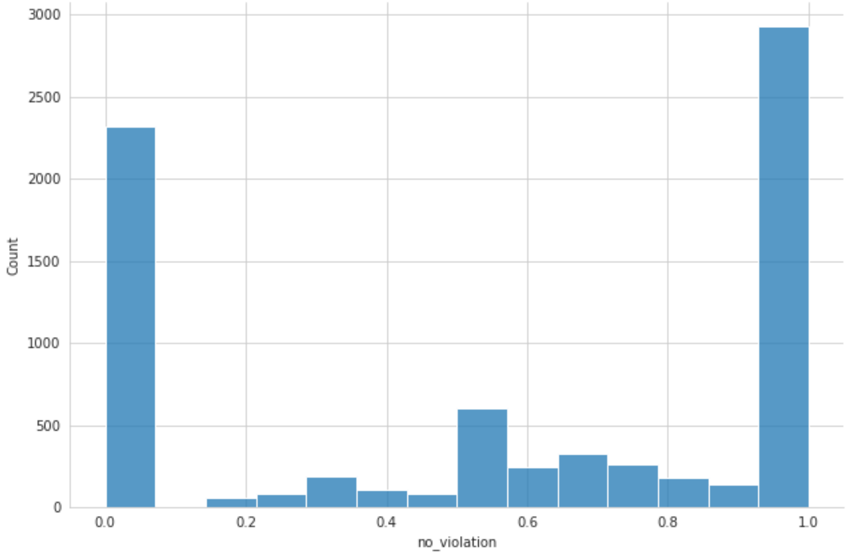


Figure 14: Distribution of the different ratios of non-violated shapes for each customer.

Some constraints or parameters that we set to simplify the analysis:

- For our specific use case at elevait, and for the model for text extraction, a threshold of 0.85 for the confidence was chosen.
- There are semantic classes for checkboxes and for fields where text is written manually. As both use different models, we focus solely on the later classes, to reduce the scope of the analysis. In the end, only 91 semantic classes were kept.
- We also only consider the 30 templates with the highest number of documents to simplify the analysis and only keep the most relevant templates.

3.6 Second analysis

The original dataset was filtered to keep only the selected templates and "numeric" semantic classes. Then, for each "template - semantic class" pair, the amount of documents where the confidence is lower than the aforementioned threshold is computed. The result is presented in Figure 17. The templates are sorted in reverse order of amount of documents: many documents match the first template, while the last templates have few representatives. We can easily observe that the first templates are over-represented compared to the lower ones.

The next step was to compare the amount of document per template and per class to the ratio of documents with low confidence (lower than the threshold). Thus, for each "template - semantic class" pair, the amount of values extracted with low confidence was divided by the total amount of values extracted for the pair. We obtain the ratio of extracted values for each class for each templates with confidence lower than the threshold. The obtained ratios are presented in Figure 18, where all values are percentages (between 0 and 100%). All values with dark purple color are at least 25%. So for the corresponding class and template, it means that at least a quarter of the documents had a low confidence on the related text field. Mainly, the class "numeric:cxYl", "numeric:cxYr", "numeric:czYl" and "numeric:czYr" (with $Y \in [1,5]$) have the most frequent low confidences over all the different templates.

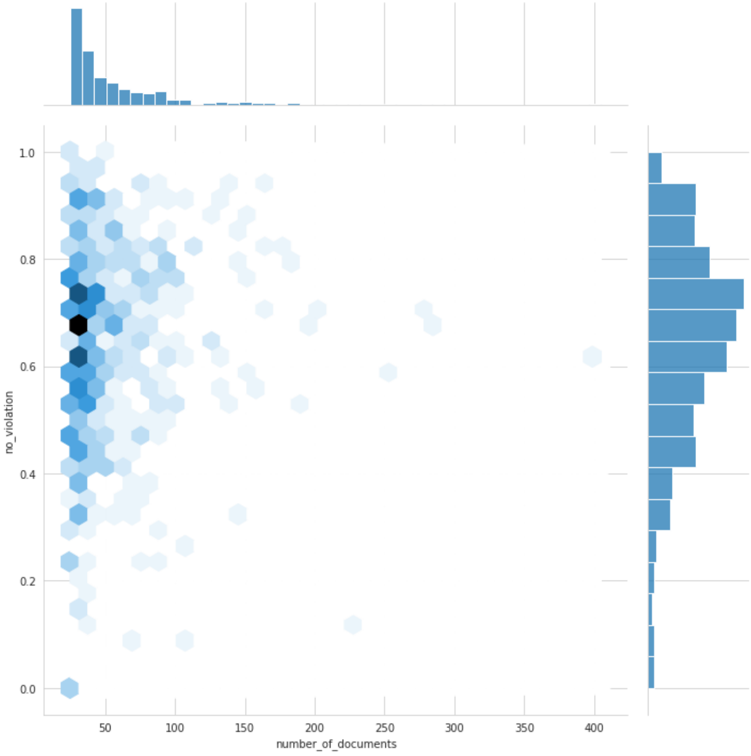


Figure 15: Distribution of the different ratios of non-violated shapes for each customer who sent more than 25 documents depending of the amount of documents they sent (with customer distribution).

3.7 Interpretation of the results

Some templates are very similar to each other. Semantic classes with the same always corresponds to the same conceptual value, even on different templates. For instance the length of the leg can be present on several forms which ask for legs and thigh measurements. The classes with a high ratio of low confidences mentioned in section 3.6 all belong to templates of forms made for hand measurements. An extract of such unfilled form can be seen on Figure 19, with the text field to be filled displayed in gray.

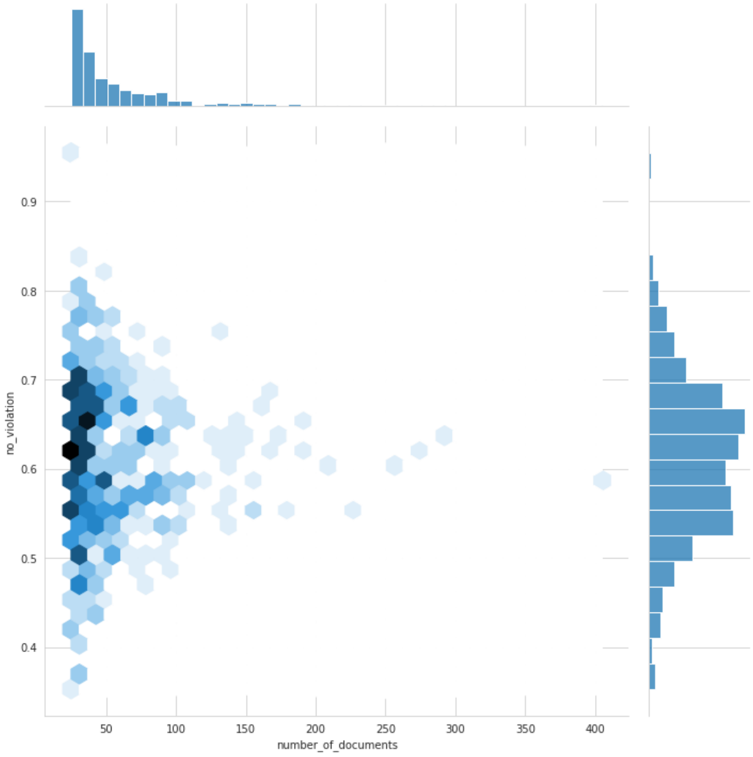
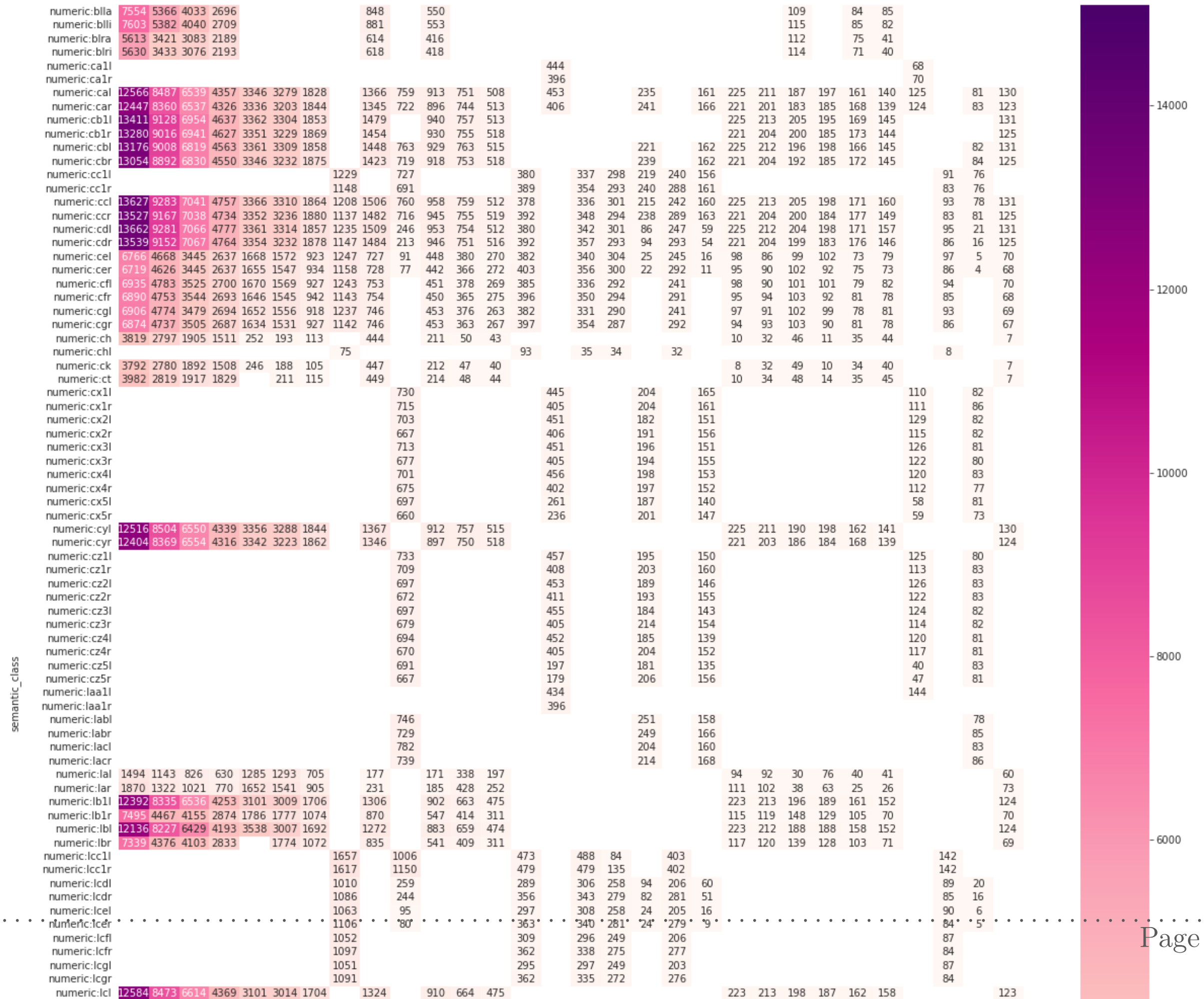


Figure 16: Distribution of the different ratios of non-violated shapes for each customer who sent more than 25 documents depending of the amount of documents they sent (with customer distribution).



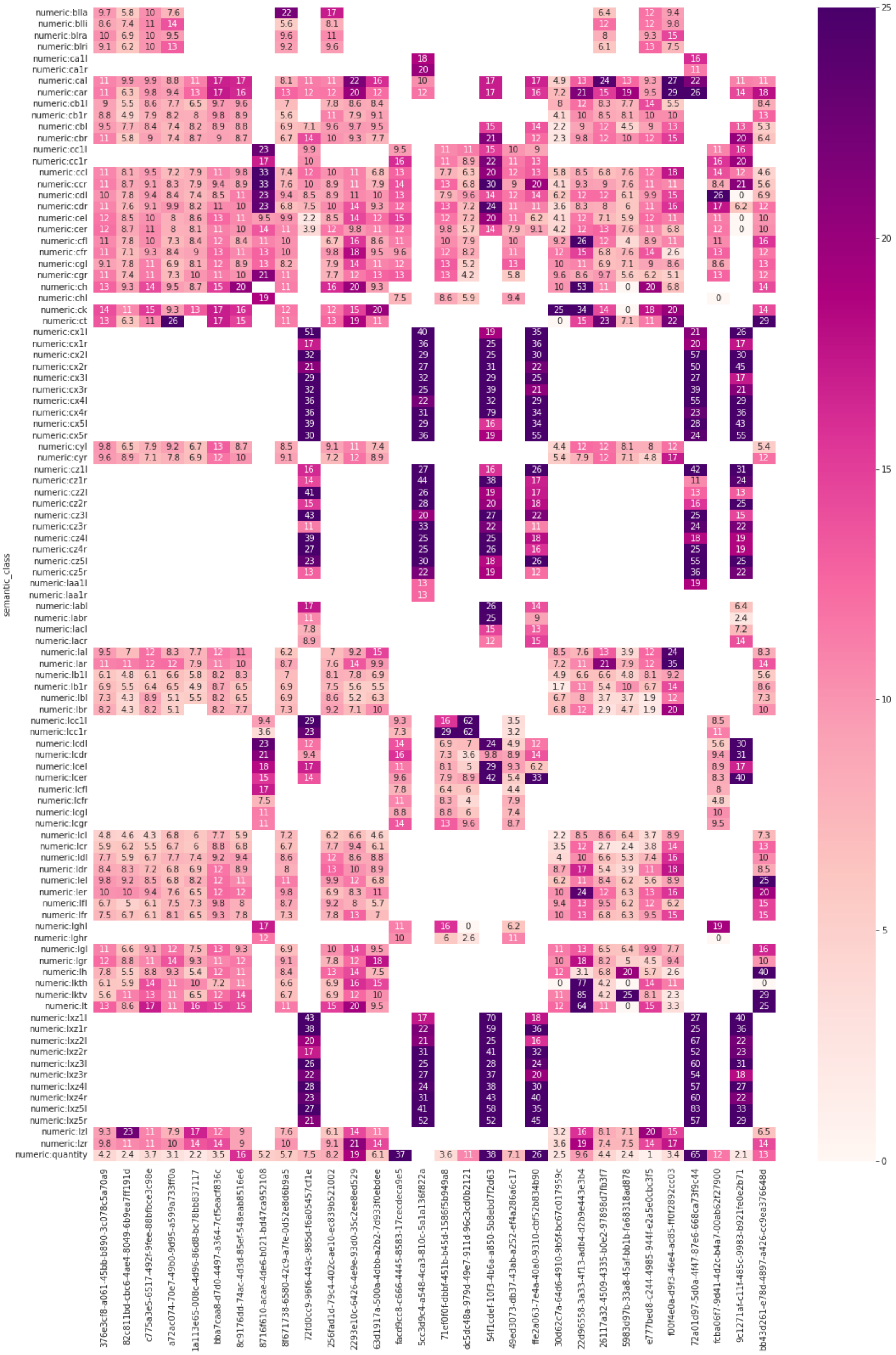


Figure 18: For the most frequent templates and each numeric class, the percentage of templates where the confidence for the value extraction for the class is lower than the threshold.

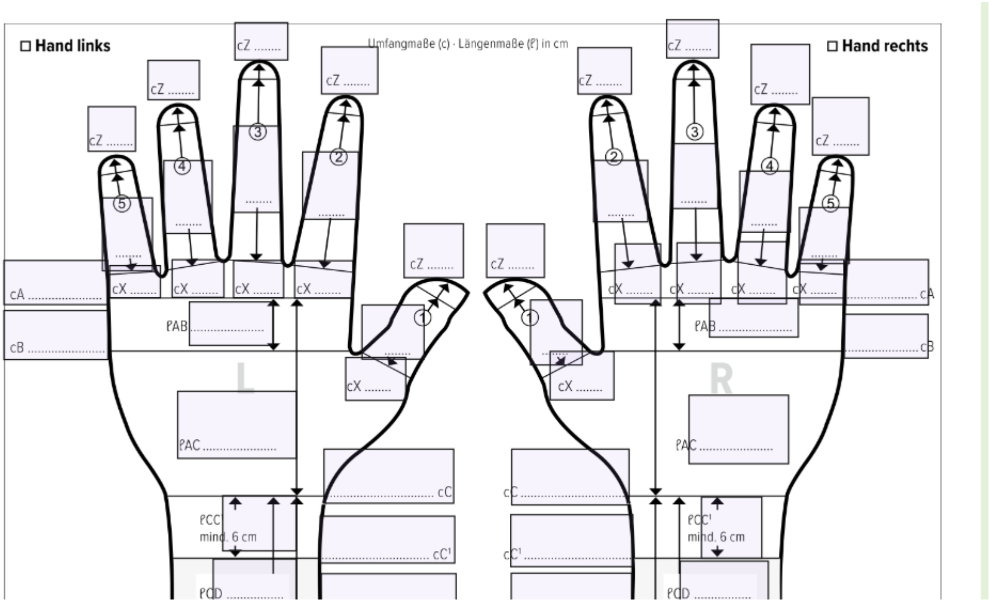


Figure 19: Cropped example of a form for hand measurement.

On the example form, some fields overlap, are quite small, or their name or what they represent may not be clear for every person. Thus, even though the confidence may be quite low, the reason is unclear: the templates may be ill-formed, the fields could be better chosen, but also the model or pre-processing could be insufficient for the extraction of the text to be optimal. The conclusion of the second analysis is that some semantic classes really have a higher probability of having low confidence, but with the current state, no actual satisfying explanation can be found. Further analysis would be necessary to find the actual cause: checking or changing the size of the fields, changing the model, or other action which would be a lot more effort and time-consuming.

3.8 Conclusion for the Use Case

In the previous sections, we defined 2 distinct hypothesis on the given dataset which are interesting for the customers of the product in:forms. They are expected to give some improvement for the automated extraction if they are true. Before developing a clustering or classification approach to be implemented within the DAIKIRI pipeline, we did in-depth statistical analysis for both ideas. Unfortunately, the outcome of the investigations show that the hypotheses are not that useful for the project:

- **Hypothesis 1**: It was not possible to identify specific classes of customers doing common errors. On one hand, the distance is small as well as the number of potential clusters, e.g., to make use of algorithms like k-means, is not obvious.
- **Hypothesis 2**: Analyzing the issues on a template or their contained fields with semantic classes support that hypothesis, however, it just underlines already known problems of small fields in a small set of templates. The customers are aware of it already. This is in the end a simple analysis where the DAIKIRI pipeline has no practical value.

Due to this considerations and the missing value in research as well as for the practical use case, we decided not to implement the use cases within the DAIKIRI pipeline.