Report: Julius Bär - Onboarding Quest

Team Based Bayes: Shakir Yousefi, Igor Pradhan, Rahul Steiger, Joel André

.

The Task

Decide on whether we accept or reject a client based on onboarding data. Binary classification for the mapping

$$\texttt{Onboarding Data} \rightarrow \{\texttt{Accept (1), Reject (0)}\}$$

We see our model as another line of defence. We assume a human in the loop can overwrite erroneous rejections, thereby making both false positives, and true negatives acceptable. Accuracy is the metric we optimize for.

Method

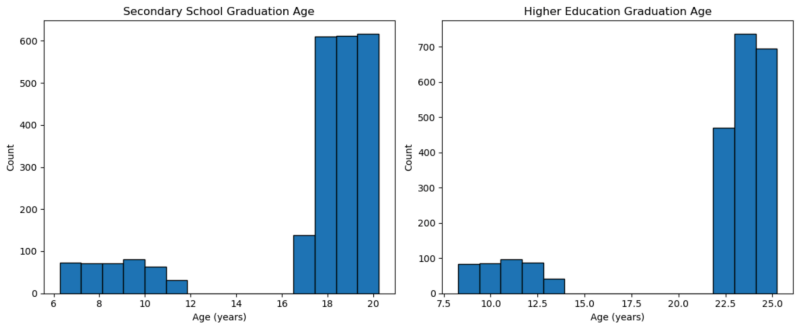
We tackled the problem in a stepwise manner, gradually increasing methodological complexity. Beginning with straightforward rule-based checks, we later introduced machine learning models to capture more subtle patterns and improve classification performance beyond what was possible through deterministic logic alone. The sections below reflect this progression.

Step 1: Tabular Consistency Checks

We initially examined tabular consistencies across the datasets 'passport', 'client_profile', and 'account_form'. Specifically, we reject onboarding in cases where fields such as first_name, middle_name, last_name do not match.

Step 2: Rule-Based Modelling and Sanity Checks

Next, we implemented rule-based modelling grounded in the structure and content of the data. A key insight was that many rejected onboarding cases involved inconsistencies in the client's timeline. For instance, we used the birth_date field in combination with the graduation_year fields from both higher_education and secondary_school to verify chronological plausibility. These sanity checks helped us catch implausible or contradictory entries that were strong indicators of rejection.



Furthermore, we assume that all submissions were made on the 1st of April 2025. If 'passport' has a 'passport_expiry_date' before this, we also reject it. Indeed, there are cases of accepted applications with an expired passport. We treat this as label noise and use this both as a rule, but also as a cleaning step. Additionally, we also check the consistency of 'marital status' by simply checking if the string is contained in the 'family background'. Initially, we tried zero-shot classification with a large-language model on marital status, but found no difference between using rule-based approaches. We also reject all submissions with empty fields with the exception of 'middle_name'.

Step 3: LLM-based data inconsistency detection.

Rule-based methods often struggle with text data, so we used LLMs to detect inconsistencies between client profiles and descriptions. We hosted a Qwen-2.5 72B model with VLLM due to limited access to proprietary vendors. Despite experimenting with prompts and input formats, the model's black-box nature and instability led to too many false positives. Given their high cost and compute demands, LLMs remain suboptimal for tasks solvable by simple rules—but improving performance and lower costs make them promising for the future.

Step 4: Lightweight ML model on samples passing simple rules

We see potential in, and tried to use (not successfully) a small ML model (e.g. random forest) on carefully engineered features, trained only on the samples the model accepts far, to discover rejections due to complex interplay of different features.

Evaluation

We structured our model as a collection of rejection rules. The model rejects if at least one rule rejects the client.

For interpretability and explainability our core model not only returns the predicted label, but on rejection also the rules that were violated, and for some rules a message that pinpoints the problem.

```
Onboarding Data → ({Accept, Reject}, [<violated_rules>], [<explanations>])
```

Metrics

Since most of our rules are coded and sensible, and we expect test data to be generated the same way as training data, overfitting is not a large concern and we evaluate on the entire dataset. Because our rules are grounded in well-defined, interpretable logic rather than arbitrary heuristics, we expect them to be able to generalize well beyond the training data. Some clients are incorrectly accepted even though their passport is expired. After correcting the labels for this we get

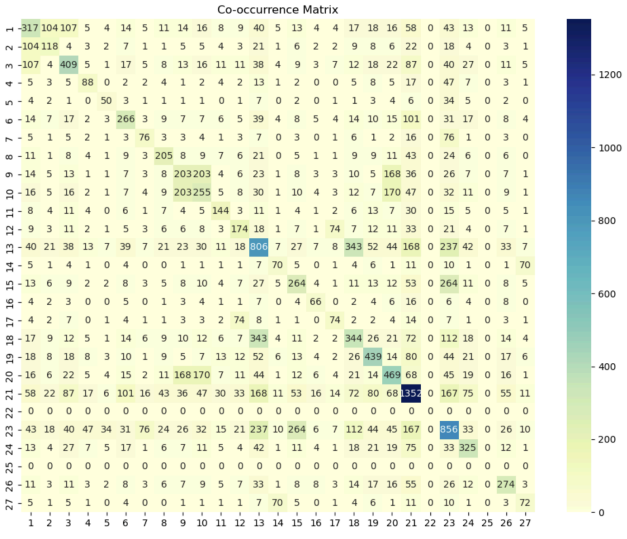
Accuracy: 90% (on the 10'000 samples)

Confusion Matrix: $\begin{bmatrix} 4122 & 1028 \\ 0 & 4850 \end{bmatrix}$

Rule Analysis

How often does a rule reject? How often is a certain rule the only rule that rejects (and thus improves our accuracy) ?

Are some rules redundant given certain other rules? All of these questions are interesting and relevant in order to decide which rules are important. We have notebooks that list these numbers for each rule. But everything can be answered with a rule rejection co-occurence matrix, so in the interest of saving space:



Rules are e.g.

1: PASSPORT_FIRST_NAME_SHOULD_MATCH_ACCOUNT_FORM_FIRST_NAME
2: PASSPORT_MIDDLE_NAME_SHOULD_MATCH_ACCOUNT_FORM_MIDDLE_NAME
13: CLIENT_PROFILE_PHONE_NUMBER_SHOULD_MATCH_ACCOUNT_FORM_PHONE_NUMBER
23: CLIENT_HAS_EMPTYFIELDS