Outline

Introduction

The loan application process is a core business of financial institutions. A well-organized process results in less delays and risks, while saving time and resources. In the past, finance managers had to check and improve the process manually. This took a lot of time and was prone to mistakes. To improve this process, a Dutch financial institution collected detailed data about its loan applications. This data comes from an online system and includes all applications made in 2016 and how they progressed until early 2017.

The loan process is divided into three main parts: the application phase, the offer phase, and the document validation and decision-making phase. In the application phase, customers submit their loan requests, and the institution waits for all necessary inputs from the customer. Once this step is complete, the offer phase begins. Here, the institution evaluates the application and makes an offer, which is sent to the customer. The customer can either accept the offer by providing documents and a signature or cancel the application. Finally, in the document validation and decision-making phase, the institution reviews the submitted documents. If everything is in order, the application is either approved or denied. If additional information is needed, the institution may request further documents. At any stage, the customer can also choose to cancel the application.

We want to make an eventually-following prediction for the *A_Cancelled* activity. This activity happens in the process after an exclusive-or-split. This process split indicates that an application is either cancelled or it is not cancelled by the user. This is a feasible task because this split happens late in the process and happens with almost every trace. The split is also well distributed, as in roughly 33% of all traces *A_Cancelled* happens.

Being able to predict this event from happening at an earlier stage of the process could prevent the bank from investing more time, energy and resources into a case that does not benefit the bank. The banks generate a revenue from the interests received on loans, so if the applications gets cancelled these are sunken costs for which the bank sees nothing in return. Moreover, predicting whether an application will be canceled also helps predict whether a loan will be successfully processed. This insight is highly relevant from a business perspective, as each approved and issued loan contributes directly to the bank’s profitability.

Business summary

The prediction model predicts whether an application will be cancelled in the future. Accurately assessing cancellation likelihood at an early stage provides significant value to the process owner. Assuming the model operates with high performance and reliability, the business gains from its implementation multiple ways. First, there is efficient resource allocation. By identifying applications at risk of cancellation early, the bank can allocate its time and personnel more effectively. They can do this by making the decision to not handle these cases all together. This is quite a drastic measure, so the bank could also decide to handle these cases with less effort by putting their focus on cases that have a higher probability of not being cancelled. Then, there are cost savings. Early prediction reduces the expenses associated with processing applications that are likely to be cancelled, which in turn lowers operational and administrative costs. On top of that, there is improved decision-making. The model provides insights that the process owner can use to make informed decisions about

which applications to prioritize, defer or terminate. Moreover, the bank can better forecast demand and manage workload. Finally, by reducing the processing of likely to be cancelled applications, the bank can offer more service to customers whose applications are progressing, improving customer satisfaction.
We aim to implement a model that predicts future *A_Cancelled* events as early as possible in the application process. It was found that after an offer is made, additional features relevant to this prediction came to light. Therefore, we have investigated the relevance of our prediction model for instances where an offer has been made. We found that the average time from the first offer created to the cancellation of an offer is 28.15 days. Meaning that, if the business owner could have predicted this cancellation, they would on average be able to save on 28.15 days of resources.

Method
There are two random forest prediction models. One predicts cancellations for applications without an offer and the other for those with an offer. The features that will be used for this prediction model are displayed in the table below. These features can be categorized into 4 categories. Event specific features (purple), case specific features (green), offer specific features (orange), and custom features (blue). The frequency of events in a prefix is computed, which is also taken into account as (event specific) features. For example, the frequency of *W_Validate application* is an important feature since it was found that the application is less likely to be cancelled after the occurrence of this event. The evidence of this feature is further elaborated in the Appendix 2.2.4.

Feature	Explanation
Prefix frequency encoding	Number of times an event has occurred in the prefix.
Application type	Indicates if the application is for a new credit or a raise for an existing credit.
Loan goal	Type of loan the customer requests.
Requested amount	Requested loan amount by applicant.
Credit score	Described whether an applicant is dependable (Gregor Scheithauer, 2017).
Monthly cost	Monthly payment amount associated with a loan offer.
Number of terms	Total number of payment installments.
Offered amount	The amount of money offered to an applicant.
First withdrawal amount	Initial amount of money disbursed or withdrawn by the loan applicant.
Total time prefix	Time between first and last event in the prefix
Prefix average time	Total time prefix divided by prefix length
Prefix length	Total number of events in the prefix
Call time	Total time an employee has been in a call with a customer

Credit score is a key offer feature (see Appendix 1.3.1), alongside monthly costs, number of terms, offered amount, and first withdrawal amount. These offer-specific features reflect the balance between applicant expectations and the bank's assessment of loan feasibility, inherently predicting application cancellations.
For case-specific features, the requested amount (detailed in Appendix 1.3.2), loan goal, and application type are important, particularly for early (pre-offer) predictions, as they are consistently available across all traces.
Custom features focus on time metrics: the total time of a prefix and the average time per activity. Slower processes are more likely to result in cancellations due to frustration or loan feasibility issues (see Appendix 1.2.3). Additionally, call time is significant, with cancelled applications averaging 158 seconds compared to 378 seconds for non-cancelled ones.

Evaluation method
For both the pre- and post-offer prefix data we trained regression models as was described above. For each of these models we report the model classification scores

such as the accuracy, recall, precision and F1-score. The support column shows on how many predictions the scores were based. We also show the ROC curves, which helps us evaluate the performance of our classification model, by assessing the trade-off between the true positive rate and the false positive rate. Finally we report on the performance results with respect to earliness. This shows how the prediction models improve given that we get further into the process. We compare the results of our model with the naive baseline prediction. The naive prediction is based on the proportion of *A_Cancelled* occurrences in the training data, where *A_Cancelled* appears in approximately 33% of cases. Therefore, the naive predictor randomly predicts that *A_Cancelled* will occur in 33% of the cases.

Preliminary results

Pre-offer Model - Accuracy: 0.59

	Precision	Recall	F1-Score	Support
0	0.68	0.73	0.70	36940
1	0.36	0.31	0.33	18124

Post-offer Model - Accuracy: 0.87

	Precision	Recall	F1-Score	Support
0	0.91	0.94	0.92	151545
1	0.58	0.50	0.54	26540

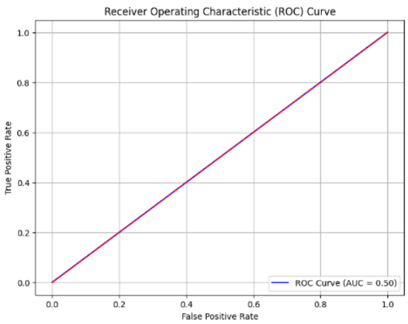
We observe that the accuracy of the post-offer model is significantly better than the baseline, however, its performance for is much better for the dominating class (0: application not cancelled) than for predicting cancelled applications. This is quite problematic given our use case.

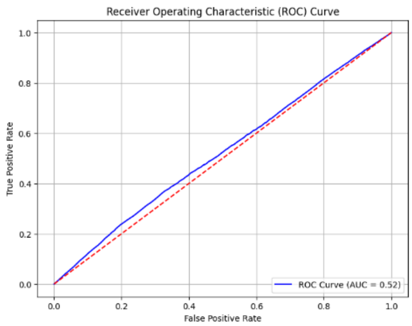
Naive Baseline - Accuracy: 0.68

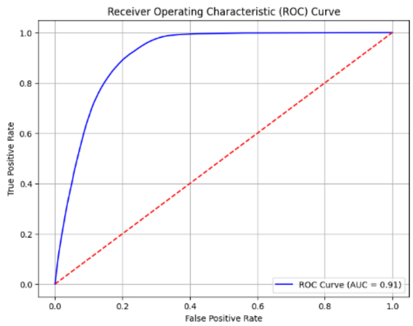
	Precision	Recall	F1-Score	Support
0	0.81	0.78	0.80	188485
1	0.19	0.22	0.20	44664

ROC curves

for the baselines, pre-offer prediction, and post-offer prediction respectively. Their corresponding areas under the curve (AUC) are 0.5, 0.52 and 0.91, where closer to 1



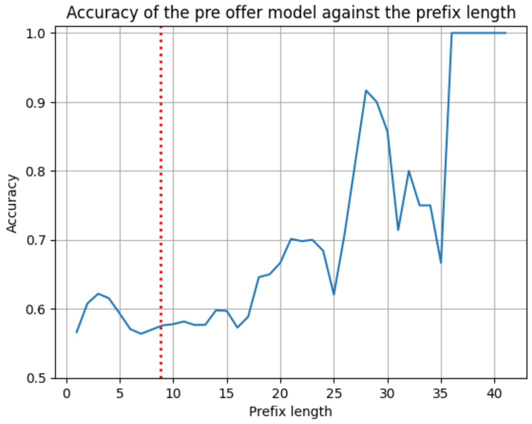


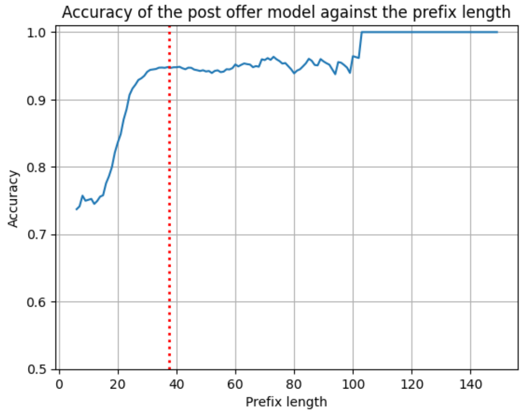


better binary classification performance.
The straight diagonal line here refers to a random model. The further the deviation from this red line, and the larger the area under the blue curve the better the model.

Earliness results

The earliness results below display the accuracy of the model against prefix length with the horizontal dashed red line showing the mean trace length (until cancellation of the application). We can see that the longer the prefix, the better the prediction. However, an almost optimal prediction is achieved around 30 activities.





Future work

We are shifting our focus from feature development into improving data filtering, class imbalance, and prediction model tuning. Because of insufficient data filtering we find in the earliness results that there are traces which have 35 activities before an offer is created. This indicates lots of seemingly outlier traces present in the data, which we plan to investigate in the coming weeks. Besides this, in the support column of the performance statistics, there is quite a significant class imbalance, which could also explain a significantly worse performance for the prediction of *A_Cancelled*. By researching ways to balance out these classes some of the bias of these models could be mitigated. Finally, we want to work on both models, to see if we can improve predictions using the features we have identified so far. The implementation is very basic so far, as it uses a random forest binary classification model with near default settings. Little hyperparameter tuning and testing has been done. We think that the foundations of our evaluation and framework are pretty good, but the models do not yet reach the performance necessary to be of substantial value to the bank. In order to reach the business value that we described, we want to make predictions that are satisfactory approximately around the time the first offer is created.

Conclusion and discussion

This project aimed to predict whether a loan application would eventually be cancelled. After frequency encoding the prefixes, features of four categories were used: trace-specific, event-specific, offer-specific and custom features. Two random forest models were created by bucketing the prefixes. One for predictions before an offer and another after the initial offer. The pre-offer-model's performed poorly. The post-offer-model performs significantly better, indicating it can distinguish cancelled cases more effectively. Evaluating earliness performance showed that predictions improve as the process progresses. Even though the post-offer model performs better, the model is significantly more accurate for prefixes of traces that are not cancelled. To fix this class bias, future work should focus on better data filtering to remove outlier traces, using methods to deal with the data imbalance, and optimizing the hyperparameter tuning of the models. Further refining the feature set could also result in better results, as the current feature set focusses more what traces are likely not to be cancelled rather than which are likely to be cancelled. Currently, the post-offer model is not effective enough to save time and money by predicting cancellations in advance. With further improvements, these predictions could become reliable enough for the bank to focus its efforts on applications that are more likely to be approved and diverging from application that are likely to get cancelled, increasing both efficiency and profit.

1 References

Gregor Scheithauer, R. H. (2017). Suggestions for Improving a Bank's Loan Application Process based on a Process Mining Analysis.

Povalyaeva, E., Khamitov, I., & Fomenko, A. (2017). *BPIC 2017: Density Analysis of the Interaction with Clients.* Moscow: National Research University Higher School of Economics.

Rodrigues, A. M. (2017). *Stairway to value: Mining a loan application process.* Rio de Janeiro: Pontifical Catholic University of Rio de Janeiro (PUC-Rio).

2 Appendix

2.1 Prediction setup & Realization of initial prediction model

Our prediction model is constructed as follows. It first receives logs in a raw format. These logs are converted to prefixes, using frequency encoding. This means for a single trace, the traces are separated into all possible prefixes of this trace, and for each unique activity the frequency of this activity occurring in the prefix is noted. This forms the foundation of the prediction input. The decision was made to go with frequency encoding, as this captures a lot of information about the prefixes, while also limiting the size and sparsity resulting dataset.

Data filtering
Any prefix containing *A_Cancelled* was removed from the dataset. That is because the event of importance had already occurred, eliminating unnecessary data. Then we filtered on uncommon end events. There are 4 common end events (*O_Cancelled, W_Call after offers, W_Call incomplete files, W_Validate application*). In addition to the common end events, there were eight rare end events identified. These uncommon events collectively account for less than 1% of the total log entries and thus were filtered out.

Bucketing
Besides the prefix encoding, we use a special bucketing strategy, which splits the data into prefixes before an offer was created (*O_Create Offer*), which we call the pre-offer-prefixes. The prefixes after an offer was created are called the post-offer-prefixes. This data is separated and fed into its own predictive model. The idea is to separate early predictions from later predictions, while keeping large enough datasets to train good models, and taking advantage of the available data in different stages of the process. This bucketing strategy is also crucial for the company as it allows for optimized predictions based on the process' phase, and leveraging newly available information. The evaluation of these separate (phase-specific) models is also more robust, specific and informative, which is crucial when you want to apply these model in the business environment.

After bucketing, the goal is to leverage as much information as possible in the feature selection procedure. In pre-offer-prefixes, we can effectively only make use of the event-specific data including prefix frequency encoding, trace-specific data, and custom features. Post-offer-prefixes can, additionally, make use of offer specific information, such as credit score, monthly costs, offered amount, first withdrawal amount, and number of terms. This data is stored concerning the last created offer in the prefix, to have the most-recent offer information as predictive features. Generally, these offer-specific features are interesting, as offers capture the dynamic between expectations of the applicant, and the estimation of the feasibility of the loan by the bank.

All chosen categorical features are one-hot-encoded, and all chosen numerical features are normalized. After this, we are ready to train the prediction models. Both models are random forest binary classification models.

Selected features
The selected features can be divided into 4 categories. Offer specific features, trace

specific features, event-specific features, and other custom features. The offer specific features are most central, as these features differentiate the pre-offer and post-offer buckets.

Credit score is a relevant offer feature, the evidence is elaborated in the Appendix 1.3.1. Generally, these offer-specific features are interesting, as offers capture the dynamic between expectations of the applicant, and the estimation of the feasibility of the loan by the bank. For this reason, we believe that monthly costs, number of terms, offered amount and first withdrawal amount are terms of the contract of which the predictive power over the cancellation of applications is natural and self-evident.

For the case-specific features we deem the requested amount, the loan goal and the application type to be of importance. The evidence of requested amount is elaborated in the Appendix 1.3.2. Although the loan goal and application type are not explicitly elaborated on with concrete evidence, it is valuable information that is generally directly available for all traces, and therefore important to include especially for early (pre-offer) predictions.

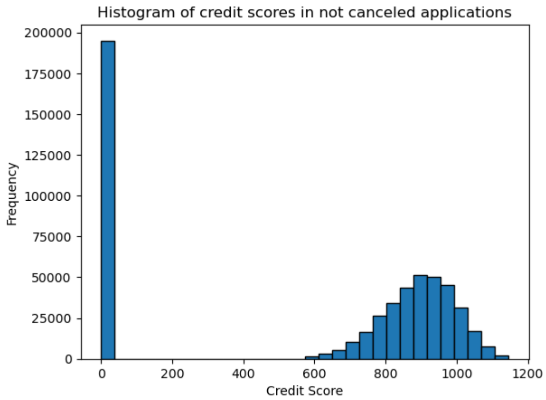
Most of the event specific features are captured within the existing prefix frequency encoding. The most important examples here are the feature which indicates occurrence of *A_Validating*, and a feature indicating the number of offers created. Although these are not explicit features (as they are part of the prefix encoding), these are deemed especially important. We base this on the very infrequent eventually follows relation between *W_Validate Application* and *A_Cancelled* expanded upon in Appendix 1.2.4. Furthermore, two studies on the same BPI 2017 dataset both show that the number of offers has an effect on the percentage of cancellations (Povalyaeva, Khamitov, & Fomenko, 2017) (Rodrigues, 2017).

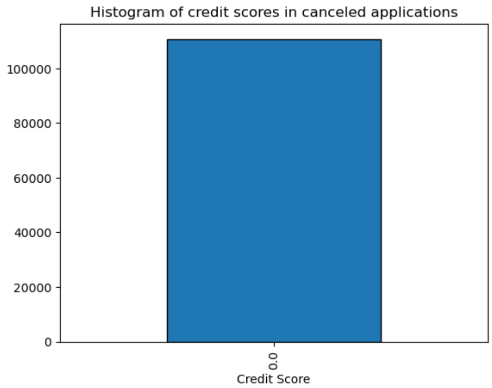
For the custom features, we focused on time. We look at the total time of a prefix and the average time per activity in the prefix (total time divided by the length of the prefix). The intuition is that "slower" processes are more likely to end up getting cancelled, either due to frustrations, or due to loan feasibility troubles. In Appendix 1.2.3, we show that traces in which the application is canceled, on average take longer per activity, and have a longer total time. Another feature we used is call time. This is the total time an employee is in a call with a customer. Calls happen in events *W_Call after offers* and *W_Call incomplete files* between the lifecycle transitions *start* and *suspend*, and *resume* and *suspend*. After the first call in a case, the cumulative call time was taken to the next events in that case. We found that the average call time for cancelled applications is 158 seconds, while the average call time for not cancelled applications is 378 seconds.

2.2 Evidence

2.2.1 Evidence credit score

The figures below show the distribution of credit score in the post-offer bucket for not cancelled and cancelled applications. Here we observe that the credit score for cancelled applications is always equal to zero. While non-cancelled applications have a credit score of 0 in 36% instances.





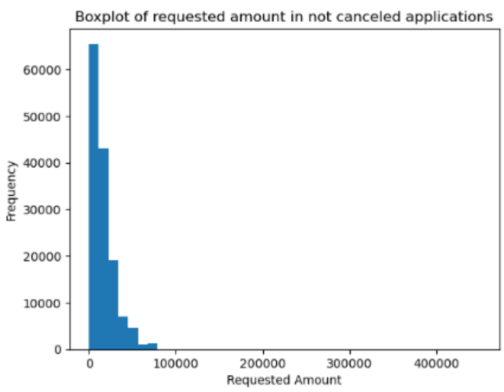
Include the visualization/table with a short description of how it was created.

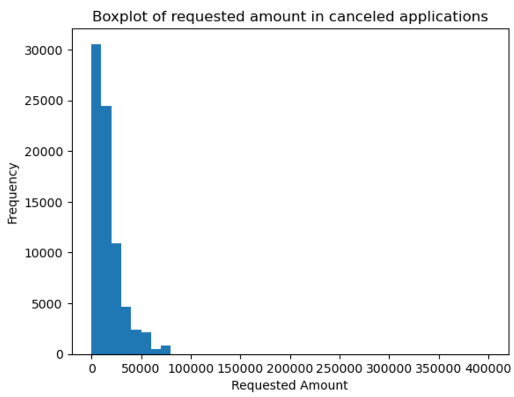
2.2.2 Evidence requested amount

The table below shows the mean and median requested amount for not cancelled and cancelled applications in the pre-offer bucket. Here we observe that the mean and median requested amount is higher for not cancelled applications. The correlation between lower requested loan amounts and higher cancellation rates might suggest that smaller loan applications may be less driven by urgent financial needs or these applications are more exploratory. Conversely larger loan requests likely reflect more serious financial commitments and greater applicant confidence, leading to lower cancellation rates.

	Mean requested amount	Median requested amount
Not canceled applications	16159	13000
Canceled applications	15616	10000

The figures below show the distribution of the requested amount in cancelled and not cancelled applications. Here we observe that the distribution of both samples look similar, they both have a lot of instances with low requested amounts, and both have a heavy right tail. Additionally, a non-parametric statistical test, specifically the Wilcoxon rank-sum test was done to determine whether there is a significant difference between the distributions of the two independent samples. The p-value was very low (p-value 4.96e-68), which suggests that the two samples do not follow the same distribution. This means that there is a significant difference between the requested amount for cancelled applications and non-cancelled applications, making this feature fit for prediction.





2.2.3 Evidence total time and average time features

The table below shows that traces containing *A_Cancelled* take longer to complete compared to traces without *A_Cancelled*. Additionally, traces with *A_Cancelled* contain fewer events. Using these statistics, we can calculate the average time per activity in hours. As shown in the table, there is a significant difference in the average time per activity between cancelled and non-cancelled applications. Therefore, a higher average time per activity strongly indicates that an application is likely to be cancelled. In the dataset, the trace length, total trace time, and average time per activity are calculated up to the current event in its specific trace.

	Mean trace length	**Mean total time of trace (in days)**	**Average time per activity (in hours)**
Not cancelled applications	44.05	17.95	9.78
Cancelled applications	26.24	29.89	27.33

2.2.4 W_Validate Application frequency

The figure below illustrates a direct follow graph on a heavily filtered dataset which only contain the activities *A_Cancelled* and *W_Validate application*. In this sense, it shows how often *A_Cancelled* and *W_Validate application* eventually follow each other in the traces of the log. As shown in the figure, *A_Cancelled* is (almost) never eventually followed by *W_Validate application* or the other way around. This indicates that the occurrence of *W_Validate application* strongly suggests that *A_Cancelled* is unlikely to eventually follow.