LTAT.05.025 – Business Process Mining: Project

Anti Alman
Anti.Alman@ut.ee

Gamal Elkoumy
Gamal.Elkoumy@ut.ee

Mahmoud Shoush
Mahmoud.Shoush@ut.ee

26.05.2021

Abstract

The increased availability of user-friendly process mining tools over the past few decades has given companies new opportunities to analyse their business processes without necessarily needing to rely on expensive technical know-how. One such tool is a web-based application Apromore, which (in conjunction with other process mining tools) is used in this project to analyse the loan application process of the European bank Dhana. This process forms the backbone of all loan services provided by Dhana and therefore it is crucial to identify any inefficiencies in this process. Furthermore, customer feedback has indicated the existence of long waiting times and non-uniform experiences in relation to the loan services provided by Dhana. This project addresses the aforementioned issues by analysing an event log of all loan applications filed with Dhana in 2016. The analysis is guided by questions provided along with the event log. The key results are provided for each question along with a detailed analysis and suggestions that will help to reduce the overall cycle time of the process and to address other inefficiencies discovered during analysis. Furthermore, the report provides additional suggestions based on prescriptive process mining and a predictive model for the loan application outcomes was developed.

1 Introduction

Analysis and optimisation of business processes has become one of the key factors contributing to the success of a company. Implementation of efficient and goal-driven business processes allows a company to gain an advantage over its competitors regardless of which services the company provides. To keep this advantage, it is important to continuously monitor the process and make adjustments. The demands and expectations of the stakeholders (both process owners and clients) change over time and as a result a process (without changes) will eventually become inefficient. Such inefficiencies can be detected and addressed by employing various process mining methods such as process discovery, conformance checking, variant analysis etc.

The purpose of this report is to employ process mining techniques in order to analyse the loan application process of the European bank Dhana. One of the core services provided by Dhana is their loan services and the loan application process forms the backbone of these services. This report is commissioned as a result of customer feedback and the churn rate of customers which indicate that the loan application process no longer meets the demands and expectations of a significant number clients.

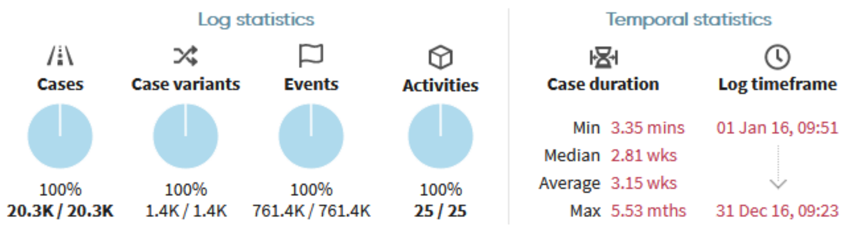


Figure 1: Overall statistics of the event log

The analysis presented in this report is based on an event log containing all loan applications filed with Dhana in 2016. Dhana has also provided 9 guiding questions as input for the analysis. The first 8 of these questions focus on the key characteristics of the process that are deemed important by the process owner (e.g. SLA compliance, repetition of activities, characteristics of cancellations, impact of application incompleteness etc.). These questions are answered mainly by using a web-based process mining application Apromore.

The 9^{th} question however allowed for free-form analysis. Based on the knowledge gathered during the analysis of the first 8 questions we decided to employ predictive and prescriptive process mining techniques. These techniques allow (respectively) to predict the outcome of the process and to identify optimisation opportunities to increase the probability of a positive outcome.

The remainder of this report is structured as follows. Section 2 and Section 3 provide a general overview of the dataset and the loan application process respectively. Section 4 addresses each of the guiding questions by highlighting the key findings, providing an in-depth analysis and summarising the related suggestions. Section 5 concludes the report by emphasizing the most relevant findings.

2 Overview of the Dataset

The analysis presented in this report is based on an event log containing all loan applications filed with Dhana in 2016. The event log contains a total of 20 343 loan applications which have been handled by 146 bank employees. For these applications, a total of 26 849 offers were created. The event log consists of 761 372 individual events. The overall statistics of the event log can be seen on Figure 1.

The event log covers the timeframe from 01.01.2016 09:51 to 31.12.2016 09:23 and does not contain any incomplete cases. The minimum case duration is 3.35 minutes, median 2.81 weeks, average 3.15 weeks and maximum 5.53 months (Figure 1).

The activities of the event log can be divided into three types: (1) activities starting with "A_" which refer to application state changes, (2) activities starting with "O_" which refer to offer state changes, and (3) activities starting with "W_" refer to workflow activities.

Workflow activity instances have start and end timestamps, meaning that the durations of these activities are available. The other two event types (Application state changes and Offer state changes) have only one timestamp corresponding to the time when an application or an offer entered the state given by the activity name. The business meaning of these activity types is further explained in Section 3.

The event log contains the following case attributes:

- ApplicationType — Always equal to "New credit";
- LoanGoal — Type of loan that was requested (11 possible string values);
- RequestedAmmount — Monetary amount that was initially requested (decimal value).

The event log also contains multiple event attributes. The relevant ones in the context of this report are[1]:

- Resource — The ID of the bank employee who performed the corresponding event;
- Accepted — Indicates if the offer was accepted or refused by the client;
- CreditScore — Credit score of the client as calculated by the bank;
- FirstWithdrawalAmount — Monetary amount to be payed out by the bank based on the given offer;
- MonthlyCost — Monthly cost of payments for the client when paying back the loan;
- NumberOfTerms — Number of terms offered by the bank for paying back the loan;
- OfferID — The ID of the offer made by the bank;
- OfferedAmount — Monetary amount offered by the bank;
- Selected — Indicates if the corresponding loan offer was selected by the client.

3 Overview of the Loan Application Process

One of the core services provided by Dhana are their loan services. These loan services in turn rely on the loan application process, the purpose of which is to handle the lifecycle of a loan application from filling out the application by the client to deciding if the loan is approved of declined by the bank. Loan payments however (both paying out the loan by the bank and paying back the loan by the customer) are out of the scope of this process. As a result the loan application process plays a crucial role in the client conversion rate in the context of the loan services and therefore it is important to detect and resolve any issues with this process.

There are three main entities in the loan application process: (1) *Loan Application* which refers to the loan application filed by a client, (2) *Loan Offer* which refers to the the loan offer made by the bank in the context of a loan application, and (3) *Workflow activity* which refers to an activity related to processing a loan application or a loan offer.

Each process instance is always related to exactly one loan application and therefore the loan application identifier is used as the case identifier throughout the entirety of this report. Loan offers are always created in the context of a loan application and multiple offers per one application are allowed, however at most one loan offer per loan application may become successful.

The workflow activities are always performed in the context of a loan application and may be specific to a single loan offer. The workflow activities are always performed by the bank and include activities such as assessing the loan application, processing the loan application, contacting the client and detecting potential fraudulent applications.

[1] A description of these attributes was not provided and therefore the meanings were deduced based on the names, values and context of the attributes.

Interestingly, creating a loan offer is not considered as a workflow activity in the event log and therefore the corresponding duration is not available for analysis. There are activities O_Create_Offer and O_Created, but the time between these is always less than 10 seconds. It would be reasonable to assume that some more noticeable amount of time is spent on creating the loan offer. Additionally, at least one loan offer is always made, however it would be reasonable to assume that at least some loan applications are rejected without making an offer (for example because of obvious fraud).

The loan application process can be seen as consisting of the following main stages:

1. Creation of the loan application by the client
2. Initial processing (including checks, corrections and initial assesment) of the loan application by the bank;
3. Creation of one or multiple loan offers by the bank and notifying the client;
4. Selection of a suitable loan offer by the client and sending the required documents to the bank;
5. Validation of the documents sent by the client to the bank in response of the loan offer (the bank may ask for additional documents or corrections);
6. Decision by the bank on whether to give out the loan to the client or not.

Overall, the loan application process is relatively well structured. The first three main steps of the process are relatively uniform for most cases, while more deviations occur after the loan offers have been created and the client has been notified. A detailed analysis of these deviations and other findings of this project are discussed in Section 4.

4 Analysis of the Loan Application Process

The following subsections address each of the guiding questions in detail. The key takeaways are provided for each question along with a detailed analysis and suggestions that will help to reduce the overall cycle time of the process and to address other inefficiencies discovered during analysis.

4.1 Task 1: SLA Compliance Analysis

Guiding question: *Are the bank's SLAs for handling loan applications met? If not, what's the frequency of SLA violations per loan type?*

The SLAs (from start to end) are as follows:

- *Car loans: 28 days*
- *Home improvement: 21 days*
- *Loan takeover: 14 days*
- *All other loans: 28 days*

Key results: There is a high frequency of SLA violations ranging from 36.6% (car loans) to 69.5% (existing loan takeovers). 28.8% of cases fall have a duration in the range of 31 to 38 days.

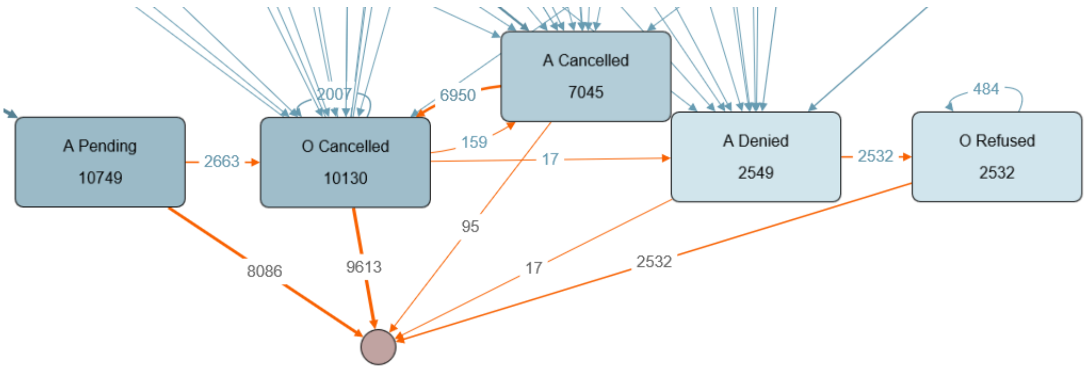


Figure 2: Exit points of the process

The duration distribution of loan cases is relatively similar regardless of the loan type.

4.1.1 Check for incomplete cases

For this task we will assume that SLA requirements are defined in terms of the total case duration. Therefore, we will first check the event log for incomplete cases by finding all the exit points of the process.

The exit points of the process (Figure 2) are: A_Pending, O_Cancelled, A_Cancelled, A_Denied, O_Refused. All of these activities refer to a final state of the loan application (activities starting with A_) or the final state of the loan offer (activities starting with O_). In all cases where the process execution ends with the final state of the loan offer it is preceded by the final state of the loan application (with practically no waiting time), therefore indicating that if a loan application reaches a final state then also all the related offers reach a final state.

Based on the above it is safe to assume that there are no incomplete cases in the given event log and no additional filtering is required.

4.1.2 Loan types

There are 11 loan types in total, with each of the the three most frequent types having a specific SLA requirement (Figure 3). Other loan types are noticeably less frequent which is most likely the reason for these types to be governed by a single SLA of "All other loans".

4.1.3 SLA Violations

For the SLA compliance we assume that the durations are inclusive (i.e. if the SLA is 28 days then the case should be completed in 28 days or less).

There is a very high frequency of SLA violations across all loan types (Table 1). Especially concerning is the fact less than one third of existing loan takeover applications are processed in time. The overall statistics for the loans of each SLA type are given on Figure 4.

A significant number of cases across all loan types have a duration between 31 to 38 days (Figure 5). This indicates that processing of the loan application is terminated after the application has been inactive for a fixed amount of time. There are 5 863 (28.8%) cases in total in this duration range.

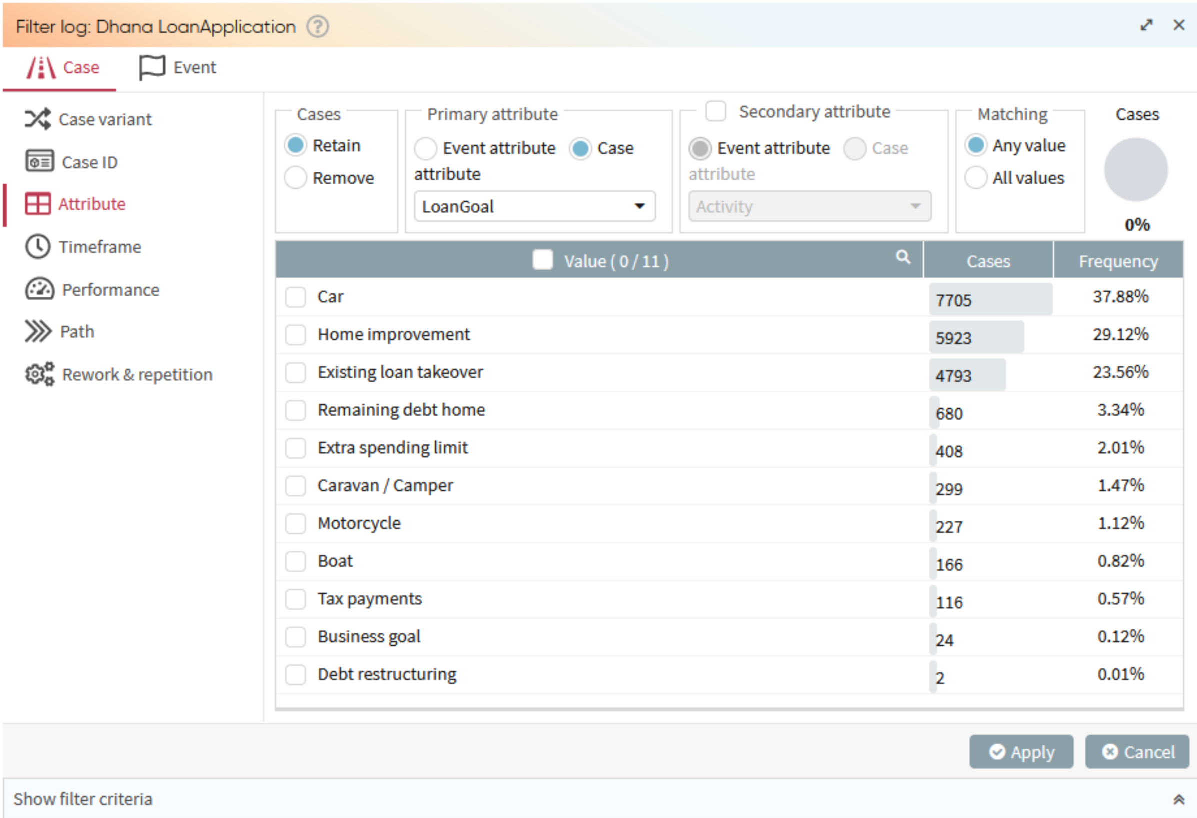
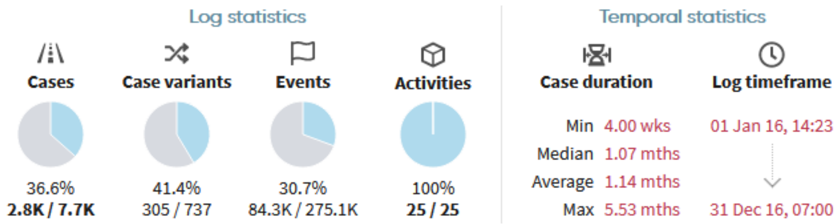


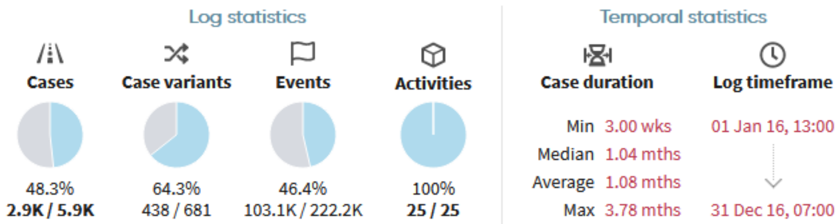
Figure 3: Loan type frequency

Table 1: Overview of the SLA violations

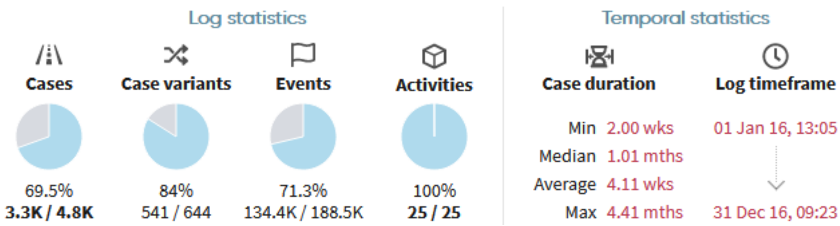
Loan type	SLA	Total	Fulfilled	Violated	Violations %
Car	28 days	7 705	4 888	2 817	36.6%
Home improvement	21 days	5 923	3 062	2 861	48.3%
Existing loan takeover	14 days	4 793	1 462	3 331	69.5%
All other	28 days	1 922	1 150	772	40.2%



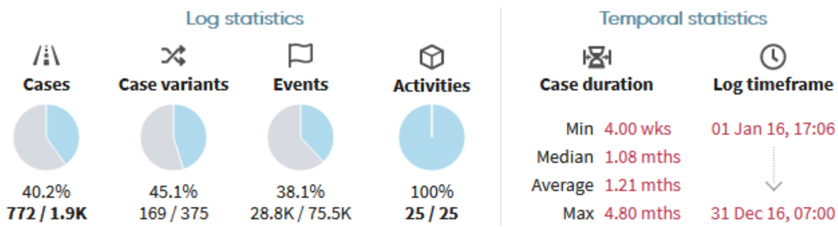
(a) Car loan cases violating the SLA



(b) Home improvement loan cases violating the SLA



(c) Existing loan takeover cases violating the SLA



(d) All other loan cases violating the SLA

Figure 4: Overall statistics for the loans of each SLA type

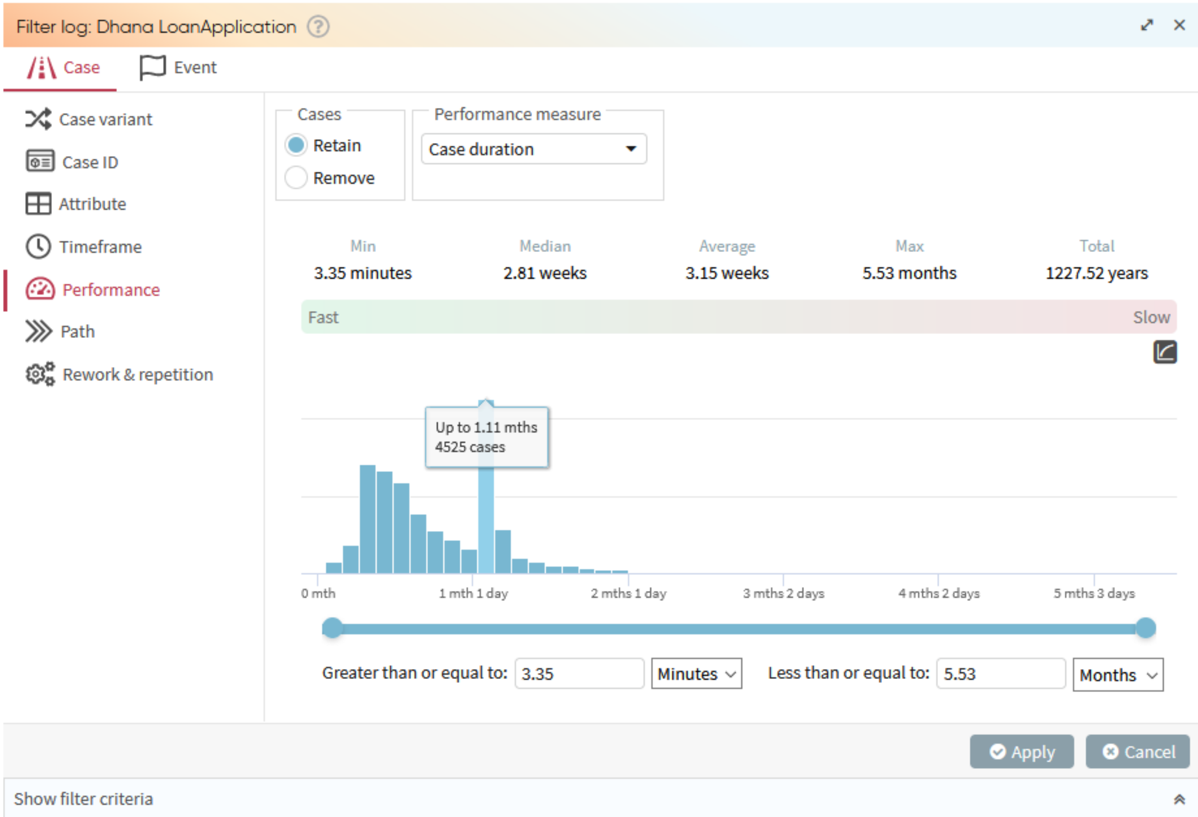


Figure 5: Duration distribution of all loan cases (range from 32 to 35 days is highlighted)

Furthermore, the duration distribution of all loan cases is relatively similar (Figure 5). This indicates a systemic issue may effect all loan types and/or that the specific SLA requirement of the loan type is not taken into account when processing the loan application.

4.1.4 Summary and Recommendations

There is an overall high frequency of SLA violations across all loan types. Car loans have the lowest frequency of violations at 36.6% while the highest frequency of violations at 69.5% was detected for existing loan takeovers (which also has the strictest SLA out of all loan types). If it is only possible to focus on a single loan type at this time then it is recommended to focus on existing loan takeovers.

However, in relation to the above recommendation it is important to emphasise that the duration distribution of of all loan cases is relatively similar regardless of the loan type. It is likely that there are common underlying causes and/or the different SLA requirements of different loan types are not enforced in practice. Therefore, it is strongly recommended to check how the SLA requirements are enforced in practice and to review or implement prioritisation rules for when specific loan applications are processed.

28.8% of cases fall have a duration in the range of 31 to 38 days. The most likely cause is an automatic inactivity threshold after which the loan application is cancelled by the bank. This may lead to a potential loss of clients as some clients may have postponed replying to the loan offer for various reasons. To counteract that potential loss of clients it is recommended to implement an (automatic) system that would send notifications to the client before the loan application is cancelled by the bank. Additionally, it is recommended to further analyse if an automatic cancellation period is necessary and if so, then is 30 days the correct threshold.

4.2 Task 2: Cycle Time Analysis

Guiding question: *The cycle time up to a milestone is the average time between the moment a case starts and the moment it reaches a given milestone. Each of the above application states is a milestone. What are the cycle times at each of the milestones of this process? Some of the milestones in this process represent points where work needs to be done by the bank. Other milestones are points where an input is needed from the customer. How much of the cycle time of this process is being spent in work done inside the bank versus how much is being spent waiting for inputs from the loan applicants. In other words, to what extent the bank is responsible for the cycle time of the process, and to what extent the loan applicant is responsible for this cycle time.*

Key results: The milestone A_Complete is reached in 1.7 days after which there is a significant jump in durations. The next milestone A_Validating is reached in 1.64 weeks from the beginning of the case. The cycle time spent inside the bank is between 4.65 to 4.79 days compared to the cycle time outside the bank which is between 1.4 weeks and 3.66 weeks.

4.2.1 Defining and examining the cycle time of different application states

To analyze the cycle time between the moment a case starts and the moment it reaches a given milestone, we first need to define the milestones we have concerning the application state. In the loan process, the start point or the first state is "A_Create_Application," which refers to the start of a case. The customer creates applications on the website or the application is created by the bank in the presence of the client. When an application is created on the website, it moves to an automatic validation step which may lead to an additional call to the client in order to request further information.

A loan application goes through a lifecycle consisting of the following states (as given in the description received from Dhana):

- **Application Created:** a new application has been created via the website;
- **Submitted:** a customer has submitted the application via the website. If the new application is created directly by the bank, this state is skipped.
- **Concept:** the application is in the concept state, which means that the customer just submitted it (or the bank started it), and a first assessment has been done automatically. An employee calls the customer to complete the application.
- **Accepted:** after the call with the customer, the application is completed and assessed again. If there is a possibility to make an offer, the state is accepted. The employee now creates one or more offers.
- **Complete:** the offers have been sent to the customer, and the bank waits for the customer to return a signed offer along with the rest of the documents (payslip, ID, etc.)
- **Validating:** the offer and documents are received and checked.
- **Incomplete:** if documents are not correct or some documents are still missing, the application state is set to 'incomplete,' which means the customers need to send in further documents.

- **Pending:** if all documents are received, and the assessment is positive, the loan is final, and the customer is paid.
- **Denied:** if somewhere in the process, the loan cannot be offered to the customer because the application does not fit the acceptance criteria, the application is declined, which results in the state 'denied.'
- **Cancelled:** if the customer never sends in their documents or calls to tell them they do not need the loan anymore, the application is canceled.

Each Application state is a milestone and the process can be analysed both in terms of the duration between the beginning of a case and a specific milestone and in terms of the time duration between different milestones. Table 2 shows the average cycle time from the beginning of a case up to a milestone. These cycle times are determined by first filtering out all events except A_Create_Application and the milestone event and then taking the average cycle time of the cases (considering only the two remaining events). Note that all loan application cases always begin with the event A_Create_Application which makes it possible to apply this type of filtering successfully.

4.2.2 Defining and examining the cycle time spent by the bank and clients

To answer the second part of the question, we first need to determine the principal components of the loan process, which correspond to the time the company spends waiting for a customer, and the time required by internal processing of requests inside the bank. The results are added to Table 2 (column "Responsible").

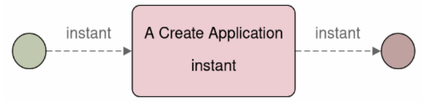
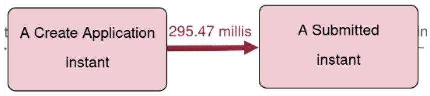
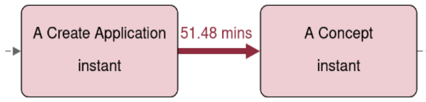
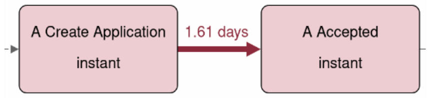
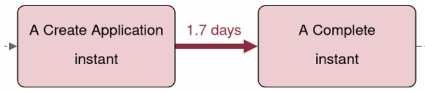
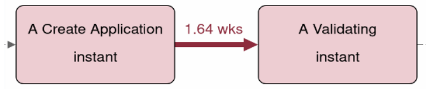
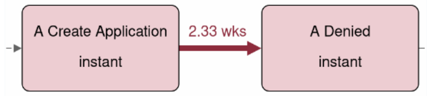
The following times were considered:

1. Time spent inside the bank (i.e. work done by the bank) when customers wait for the bank to reply:
   - from the completion of the automatic first assessment (event A_Concept) until completion of the application (event A_Complete);
   - from when the client has sent the required documents to the bank (event A_Validating) until making a final decision on giving out the loan (events A_Pending and A_Denied)[2].
2. Time spent at the customer side (i.e. work done by the client) when the bank waits for the customer to reply:
   - from the beginning of creating an application (the beginning of the process) until the automatic first assessment is completed (event A_Concept);
   - from completion of the application (event A_Complete) until the completion of reviewing the offer and sending the required documents to the bank (event A_Validating);
   - from when the client has sent the required documents to the bank (event A_Validating) until cancelling the loan application (events A_Cancelled)[3].

[2]The duration from A_Validating to A_Incomplete is not considered separately as this duration is included in the duration from A_Validating to A_Pending or A_Denied

[3]Cancelling is considered as time on the customer side even in case of automatic cancellations by the bank as it is still technically time during which the bank is waiting on the client to take action

Table 2: Average cycle time from the beginning of a case up to a milestone

Milestone activity	Avg. Cycle time	Responsible	Apromore screenshot
A_Create_Application	instant	Client	
A_Submitted	295.47 Millis	Client	
A_Concept	51.48 minutes	Client	
A_Accepted	1.61 days	Bank	
A_Complete	1.7 days	Bank	
A_Validating	1.64 weeks	Client	
A_Incomplete	1.91 weeks	Bank	
A_Pending	2.59 weeks	Bank	
A_Denied	2.33 weeks	Bank	
A_Cancelled	4.28 weeks	Client	

(a) A_Concept ⟶ A_Complete

(b) A_Validating ⟶ A_Pending

(c) A_Validating ⟶ A_Denied

Figure 6: Estimation of average cycle time spent inside the bank

4.2.3 Time spent inside the bank: a customer is waiting

In this section, we will provide an estimation on how much of total cycle time is spent inside the bank (i.e. a customer is waiting for the bank to execute some activities). *(1)* Once the customer submitted an application and the automatic first assessment is completed (event A_Concept), he/she should wait for the bank's response which is sent by the time when A_Complete occurs. The average cycle time between these two events is 1.67 days (Figure 6a). *(2)* Once the customer has reviewed the offers and sent the required documents to the bank (event A_Validating) the bank must either confirm the loan (event A_Pending) or deny the loan (event A_Denied). The average cycle times are respectively 3.3 days (Figure 6b) and 2.98 days (Figure 6c). Therefore, the total average cycle time spent inside the bank is 4.97 days for confirmed loans and 4.65 days for denied loans.

4.2.4 Time spent outside the bank: the bank is waiting

In this section, we will provide an estimation on how much of total cycle time is spent outside the bank (i.e. the bank is waiting for the customer to execute some activities). *(1)* From the beginning of the case the bank is first waiting for the application to be created and the first automatic assessment to be completed (event A_Concept). The average cycle time until this event is 51.48 minutes (Figure 7a). *(2)* Once the application is completed (event A_Complete) the customer must review the offers and send the required documents to the bank (event A_Validating). The average cycle time between these two events is 1.4 weeks (Figure 7b). *(3)* Once the client has sent the required documents to the bank (event A_Validating) the client may cancel the loan application (event A_Cancelled). The average cycle time between these two events is 2.26 weeks (Figure 7c). The time spent on *(1)* is negligible compared to the time spent on *(2)* and *(3)*. Therefore, in the cycle time spent outside the bank is 1.4 weeks for cases during which the loan application is not cancelled and 3.66 weeks for cases during which the loan application is cancelled.

4.2.5 Summary and Recommendations

The milestone A_Complete is reached in 1.7 days after which there is a significant jump in durations. The next milestone that comes after A_Complete is A_Validating, which is reached on average in 1.64 weeks from the beginning of the case. This indicates that the first milestones of the process are relatively efficient and therefore optimisation efforts should first prioritise activities coming after the application is completed and the initial offers are made.

Most of the cycle time of the loan application process is spent outside the bank. The cycle time outside the bank which is between 1.4 weeks and 3.66 weeks, while the cycle time spent

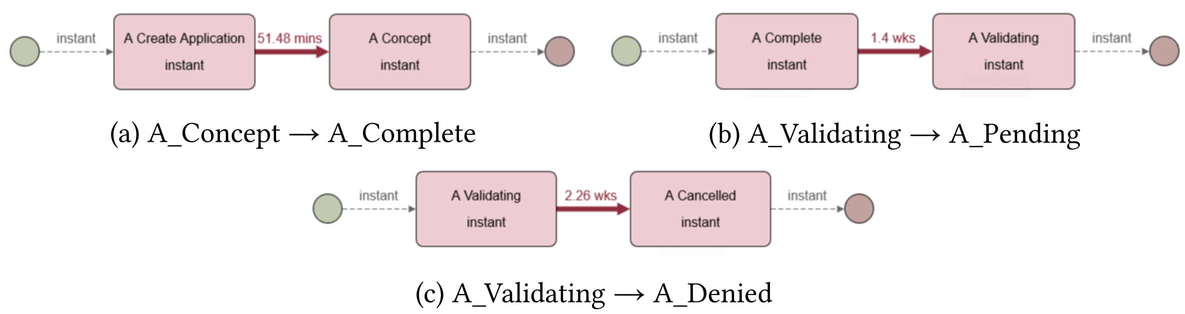


Figure 7: Estimation of average cycle time spent outside the bank

inside the bank is between 4.65 to 4.79 days. While the bank does not have direct control over the efficiency of the activities performed outside the bank, it may still be possible to effect these activities indirectly. Similarly to Section 4.1 it is recommended to implement an (automatic) system that would send notifications to the client. By choosing a reasonable interval of notifications (for example once or twice every week) it may be possible to influence the clients to be more active.

The cycle time both inside and outside the bank varies based on the final status of the loan application. The cycle time spent outside the bank is 1.4 weeks for cases during which the loan application is not cancelled and 3.66 weeks for cases during which the loan application is cancelled. This is most likely the cause of client inactivity and an automatic cancellation threshold as also speculated in Section 4.1.

4.3 Task 3: Rework Analysis

Guiding question: *Is there any rework loop in this process? If so, what is its impact of each rework loop on the cycle time (i.e. by how much does the occurrence of the rework loop increases the process)? Note: here we define a rework as one or a sequence of activities that occurs twice or more in the same case.*

Key results: There are 12 rework loops in this process, half of which are self-loops. The most significant rework loop is O_Create_Offer ⟶ O_created ⟶ O_sent_(mail_and_online) ⟶ W_call_after_offers ⟶ A_complete which occurs is 12.6% of cases. The cycle time of cases with this rework loop is on average 1 week longer than cases without this rework loop (4.09 and 3.01 weeks respectively). Other rework loop (while still relevant) are somewhat less significant.

4.3.1 Rework loops

The average cycle time of the event log is 3.15 weeks (Section 2). Below, we list the frequent rework loops and their effect on the cycle time by comparing cases that contain the given rework loop to cases that do not contain the same rework loop. This comparison gives a better overview that just comparing against the average cycle time as the is skewed by the existance of the given rework loop.

The loan applications event log contains the following rework loops:

1. W_Validate_Application ⟶ A_Validating ⟶ O_Returned ⟶ W_Call_incomplete_files ⟶ A_Incomplete. Shown in Figure 8.

This loop happens in 6561 cases (32.3%). The average cycle time for those cases is 2.87 weeks. The average cycle time of the event log without this loop is 3.28 weeks.

2. W_Validate_Application ⟶ A_Validating ⟶ O_Returned ⟶ W_Assess_Potential_Fraud. Shown in Figure 9.

   This loop happens in 79 cases (0.4%). The average cycle time for those cases is 3.07 weeks. The average cycle time of the event log without this loop is 3.15 weeks. It does not have a significant effect because a small amount of cases go through the loop.

3. O_Create_Offer⟶ O_created ⟶ O_sent_(mail_and_online) ⟶ W_call_after_offers ⟶ A_complete. Shown in Figure 10.

   This loop happens in 2560 cases (12.6% of the cases). The average cycle time for those cases is 4.09 weeks. The average cycle time of the event log without this loop is 3.01 weeks.

4. W_validate_Application ⟶ A_Validating ⟶ W_Call_incomplete_files ⟶ A_incomplete. Shown in Figure 11.

   This loop happens in 3565 cases (17.5% of the cases). The average cycle time for those cases is 3.41 weeks. The average cycle time of the event log without this loop is 3.09 weeks.

5. W_validate_application ⟶ A_validating ⟶ O_returned ⟶ W_assess_potential_fraud. Shown in Figure 12.

   This loop happens in 79 cases (0.4% of the cases). The average cycle time for those cases is 3.07 weeks. The average cycle time of the event log without this loop is 3.15 weeks.

6. O_create_offer ⟶ O_created. Shown in Figure 13.

   This loop happens in 2146 cases (10.5% of the cases). The average cycle time for those cases is 3.42 weeks. The average cycle time of the event log without this loop is 3.11 weeks.

7. Self Loop: O_sent_(mail_and_online). Shown in Figure 14a.

   This loop happens in 1823 cases (9% of the cases). The average cycle time for those cases is 3.46 weeks. The average cycle time of the event log without this loop is 3.12 weeks.

8. Self Loop: O_cancelled. Shown in Figure 14b.

   This loop happens in 2007 cases (9.9% of the cases). The average cycle time for those cases is 4.64 weeks. The average cycle time of the event log without this loop is 2.98 weeks.

9. Self Loop: O_refused. Shown in Figure 14c.

   This loop happens in 484 cases (2.4% of the cases). The average cycle time for those cases is 2.84 weeks. The average cycle time of the event log without this loop is 3.15 weeks.

10. Self Loop: W_assess_potential_fraud. Shown in Figure 14d.

    This loop happens in 18 cases (0.1% of the cases). The average cycle time for those cases is 3.85 weeks. The average cycle time of the event log without this loop is 3.15 weeks.

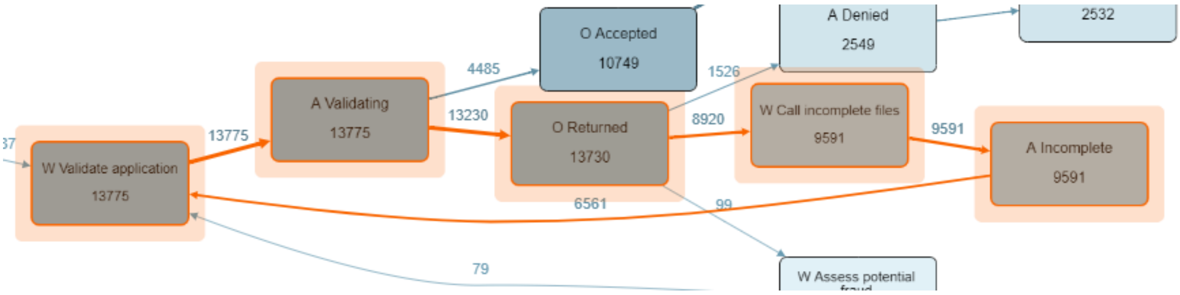


Figure 8: Loop 1

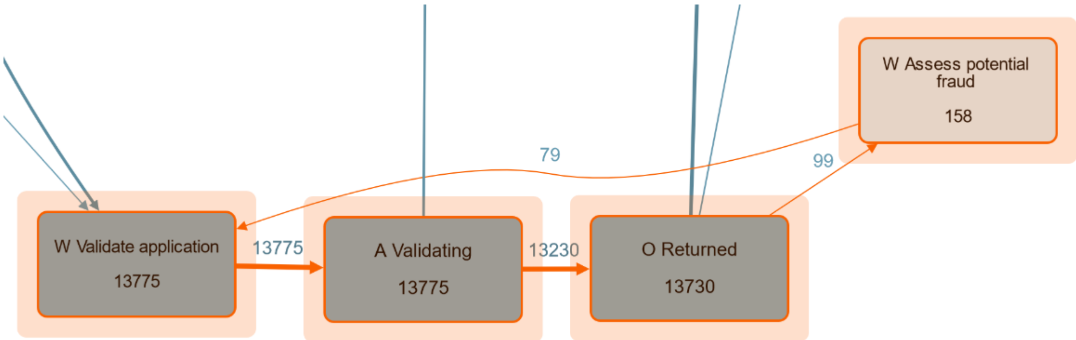


Figure 9: Loop 2

11. Self Loop: W_handle_leads. Shown in Figure 14e.

    This loop happens in 24 cases (0.1% of the cases). The average cycle time for those cases is 3.31 weeks. The average cycle time of the event log without this loop is 3.15 weeks.

12. Self Loop: W_complete_application. Shown in Figure 14f.

    This loop happens in 6 cases (0.03% of the cases). The average cycle time for those cases is 1.4 weeks. The average cycle time of the event log without this loop is 3.15 weeks.

4.3.2 Summary and Recommendations

Table 3 summaries the analysis. We conclude that the most significant rework loops are:

- O_Create_Offer ⟶ O_created ⟶ O_sent_(mail_and_online) ⟶ W_call_after_offers ⟶ A_complete.
- Self loop: O_cancelled

The first loop occurs in 12.6% of the cases and removal of that loop reduces the average cycle

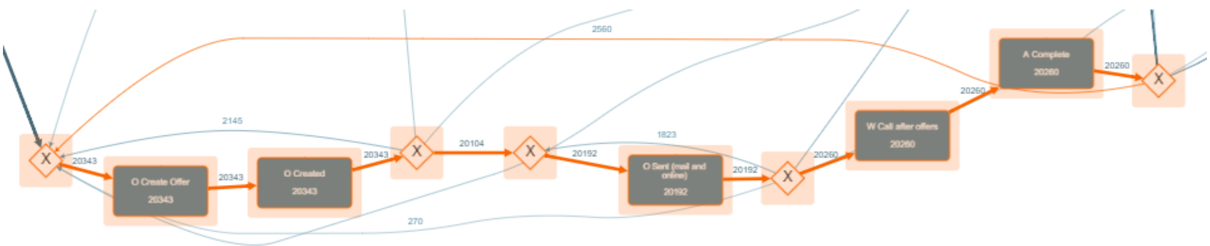


Figure 10: Loop 3

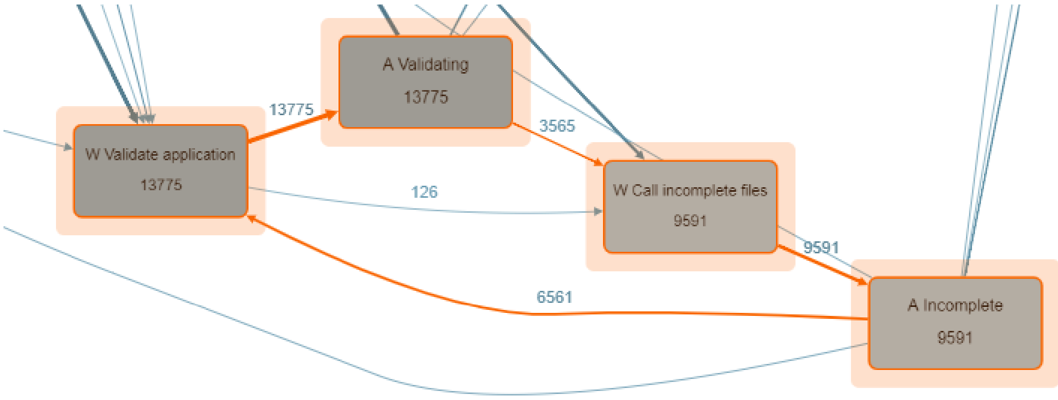


Figure 11: Loop 4

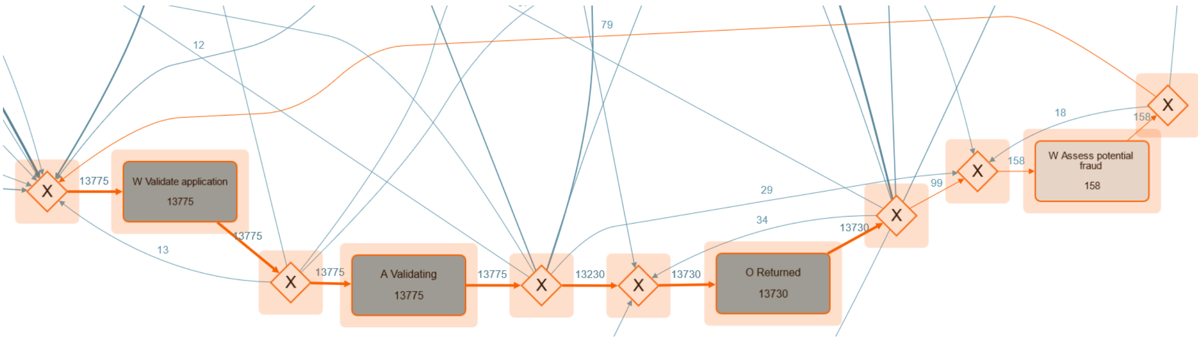


Figure 12: Loop 5

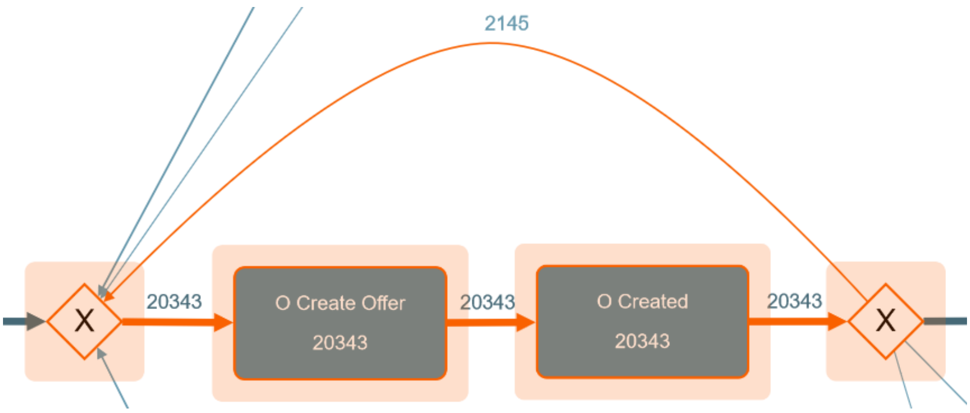
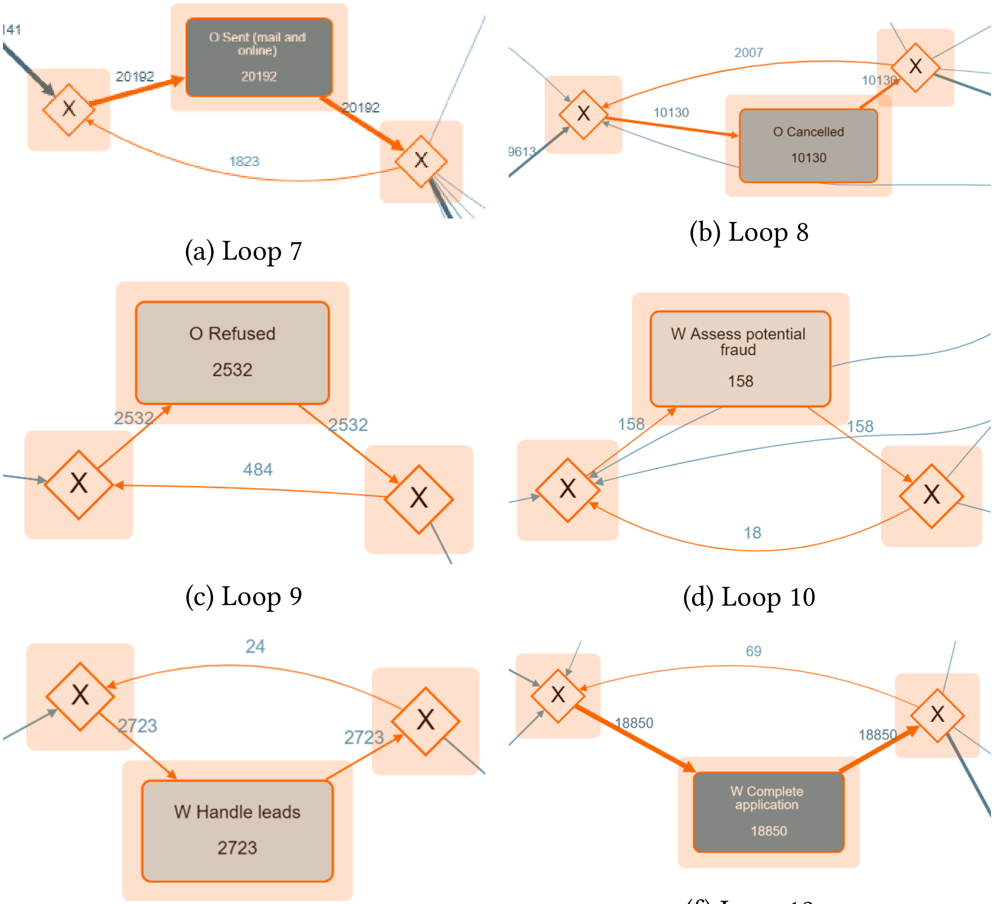


Figure 13: Loop 6



(a) Loop 7

(b) Loop 8

(c) Loop 9

(d) Loop 10

(e) Loop 11

(f) Loop 12

Figure 14: Self loops of the process

Table 3: Summary of the identified rework loops

Loop Activities	Number of Cases	Cycle time with loop	Cycle time without loop
W_Validate_Application, A_Validating, O_Returned, W_Call_incomplete_files, A_Incomplete	6561 cases (32.3%)	2.87 weeks	3.28 weeks
W_Validate_Application, A_Validating, O_Returned, W_Assess_Potential_Fraud	79 cases (0.4%)	3.07 weeks	3.15 weeks
O_Create_Offer, O_created, O_sent_(mail_and_online), W_call_after_offers, A_complete	2560 cases (12.6%)	4.09 weeks	3.01 weeks
W_validate_Application, A_Validating, W_Call_incomplete_files, A_incomplete	3565 cases (17.5%)	3.41 weeks	3.09 weeks
W_validate_application, A_validating, O_returned, W_assess_potential_fraud	79 cases (0.4%)	3.07 weeks	3.15 weeks
O_create_offer, O_created	2146 cases (10.5%)	3.42 weeks	3.11 weeks
O_sent_(mail_and_online)	1823cases (9%)	3.46 weeks	3.12 weeks
O_cancelled	2007 cases (9.9%)	4.64 weeks	2.98 weeks
O_refused	484 cases (2.4%)	2.84 weeks	3.15 weeks
W_assess_potential_fraud	18 cases (0.1%)	3.85 weeks	3.15 weeks
W_handle_leads	24 cases (0.1%)	3.31 weeks	3.15 weeks
W_complete_application	6 cases (0.03%)	1.4 weeks	3.15 weeks

time of the process to 3.01 weeks which is especially significant when compared the cycle time of 4.09 weeks of cases containing that same loop. Within these cases, 90% of the loan applications are cancelled and 10% get refused by the bank. An early detection and mitigation of this behavior can reduce the average cycle time of the process and can have a positive affect on the ratio of approved loan applications.

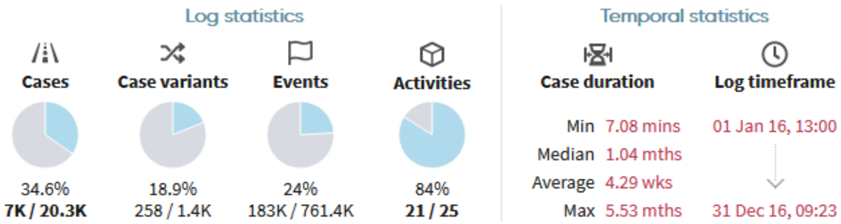
The second loop occurs in 9.9% of the cases and the average cycle time of the cases without this loop is 2.98 weeks. The bottleneck of the cases with that loop occurs with the activity A_Cancelled, because it is on average executed 4 weeks after the start of the case. The activity A_Cancelled happens when "Client declined the credit offer, didn't send the documents or is out of reach for 30 days". Reducing the inactivity threshold would also reduce the average cycle time of the entire process.

The removal of only the above two loops reduces the average cycle time to 2.95 weeks (i.e. reduction by 1.4 days (6.35%) of the average cycle time).

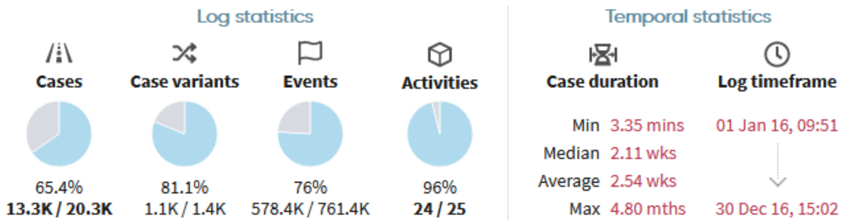
4.4 Task 4: Analysis of Cancellations

Guiding question: *How many loan applications are cancelled? What are the characteristics of the loan applications being cancelled? Is there any common pattern among them?*

Key results: There are 7045 cases where the loan application is eventually cancelled. The average cycle time of these cases is 4.29 weeks which is significantly higher than both the overall average cycle time (3.15 weeks) and the average cycle time of cases where the loan application



(a) Cases containing A_Cancelled



(b) Cases not containing A_Cancelled

Figure 15: Statistics of cases where the application is cancelled (15a) and cases where the loan application is not cancelled (15b)

is not cancelled (2.54 weeks). There is far more variance after the activity A_Complete in cases where the loan application is not cancelled and in some outlier cases there are noticeable waiting times of multiple days or even weeks after the activities A_Accepted and O_Created. The activity W_Validate_application occurs in 99.2% of cases where the application is not cancelled and only 8.29% of cases where the application is cancelled.

4.4.1 Cancellation statistics

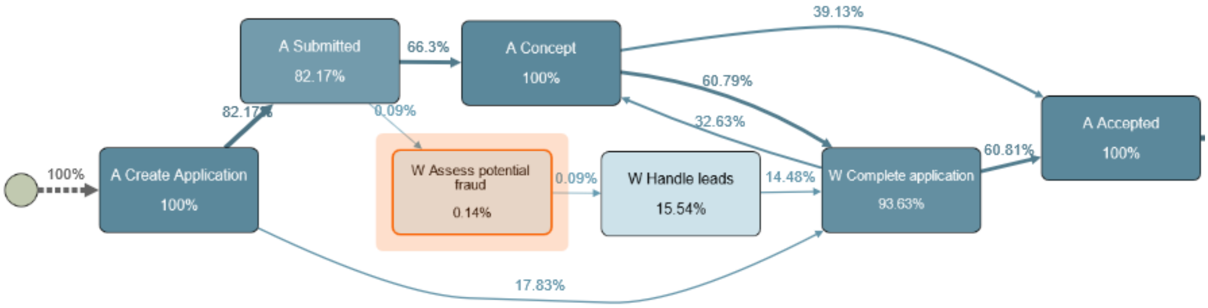
There are 7045 cases where the loan application is eventually cancelled. These cases cover 18.9% of all case variants and 34.6% of all cases (Figure 15a). There are 13298 cases where the loan application is not cancelled (including both the cases where the loan is granted and the cases where the loan is denied). These cases cover 81.1% of all case variants and 65.4% of all cases (Figure 15b).

Based on the statistics (Figure 15) we can see a significant difference in both median and average cycle times. The median cycle time for cases where the application is cancelled is 1.04 months (4.29 weeks average) while for all the other cases it is 2.11 weeks (2.54 weeks average). This may be caused by automatic cancellation of some loan applications after a period of inactivity, as was also indicated in Section 4.1.

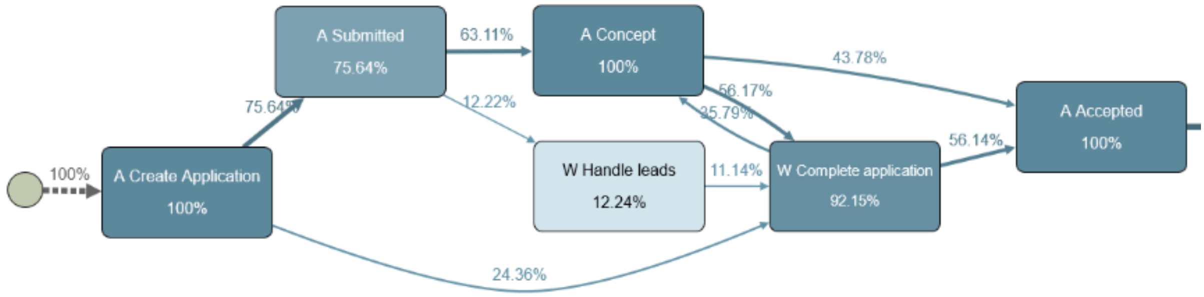
The following activities never occur in cases where the loan application is cancelled: A_Denied, A_Pending, O_Accepted, O_Refused. The first two are expected as they are the final states of the loan application. However, the O_Accepted, O_Refused are interesting, because the lack of these activities most likely indicates that no feedback was received from the customer after a loan offer was made.

4.4.2 Relative case frequencies from the start of the case to A_Accepted

The overall control flow until the activity A_Accepted is similar in both sets of cases (Figure 16). The main noticeable difference is the occurrence of W_Assess_potential_fraud which appears at the beginning of the process map only in cases where the application is eventually cancelled.



(a) Cases containing A_Cancelled



(b) Cases not containing A_Cancelled

Figure 16: Process map from the start of the case to A_Accepted with relative case frequency overlay

The same activity also occurs in cases where the application was not cancelled, however in that case it is in the latter part of the process control flow.

In terms of duration most of the differences are also minor with the only major difference being again W_Assess_potential_fraud which has a median duration of 2.64 hours in cases where the application was cancelled compared to 18.63 hours in cases where the application was not cancelled.

4.4.3 Relative case frequencies from A_Accepted to A_Complete

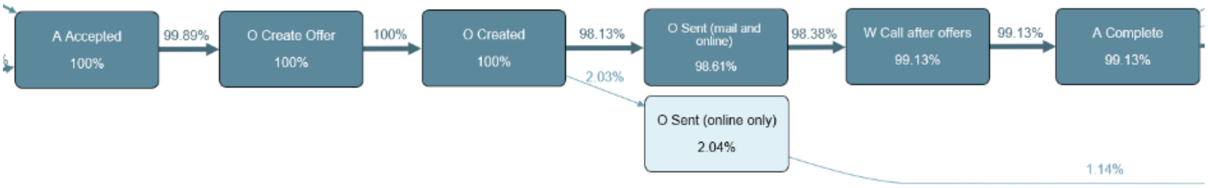
There are no noticeable differences in the order of activities in the control flow from the activity A_Accepted to the activity A_Complete (Figure 17). The only minor difference is with the activity O_Sent_(online_only) which occurs slightly more frequently in cases where the loan application was not cancelled.

The durations are similar between the two sets of cases with the only noticeable difference being in the average time between the activities O_Created and O_Sent_(online_only). The average time between these activities is 1.92 hours in cases where the loan application was not cancelled and 23.88 minutes in other cases.

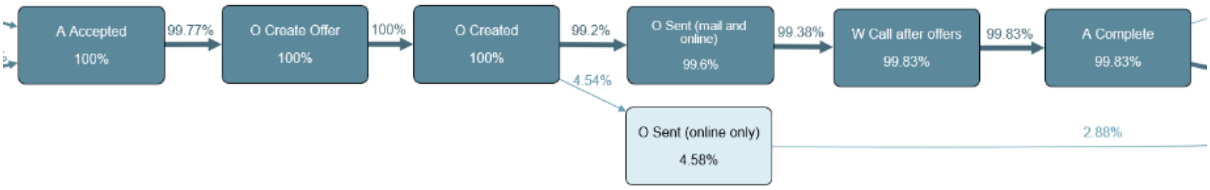
In some outlier cases there are significant delays after the activities A_Accepted and O_Created. The maximum wait time after these activities can be multiple days or even weeks regardless of the final state of the loan application reached by the end of the process.

4.4.4 Relative case frequencies from A_Complete to the end of the case

The major control flow differences occur starting from the activity A_Complete (Figure 18). The most noticeable difference is the relatively simple control flow in cases where the loan



(a) Cases containing A_Cancelled



(b) Cases not containing A_Cancelled

Figure 17: Process map from A_Accepted to A_Complete with relative case frequency overlay

application is eventually cancelled compared to the more complex control flow of other cases.

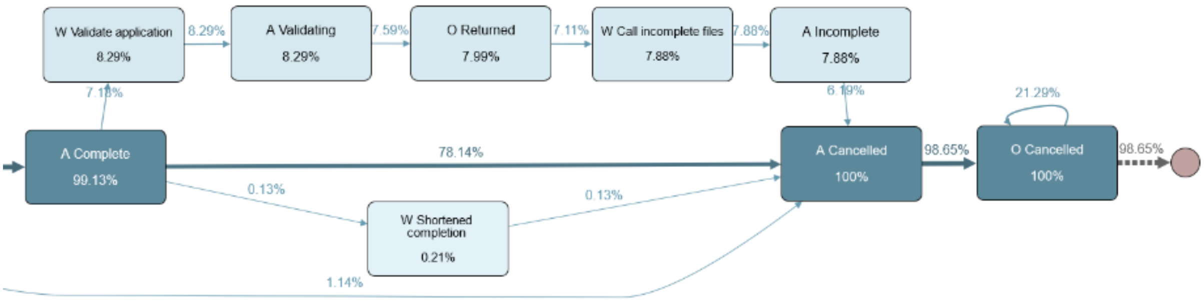
If the loan application is cancelled then in the majority of cases the cancellation occurs after the activity A_Complete without any other events occurring in-between (Figure 18a). The activity A_Complete corresponds to the offer(s) being completed and sent to the client for review and feedback. It seems that client's feedback often does not arrive in such cases, because the activity A_Cancelled refers to the bank cancelling the application (possibly due to inactivity as noted previously).

There are also two other less frequent pathways visible in case of cancellations. One of these consists of only the activity W_Shortend_completion. The second is a longer chain of activities starting with the activity W_Validate_application. The activity W_Shortend_completion has a similar frequency in both. But the longer chain starting with W_Validate_application occurs in 99.2% of cases where the application is not cancelled and only 8.29% of cases where the application is cancelled.

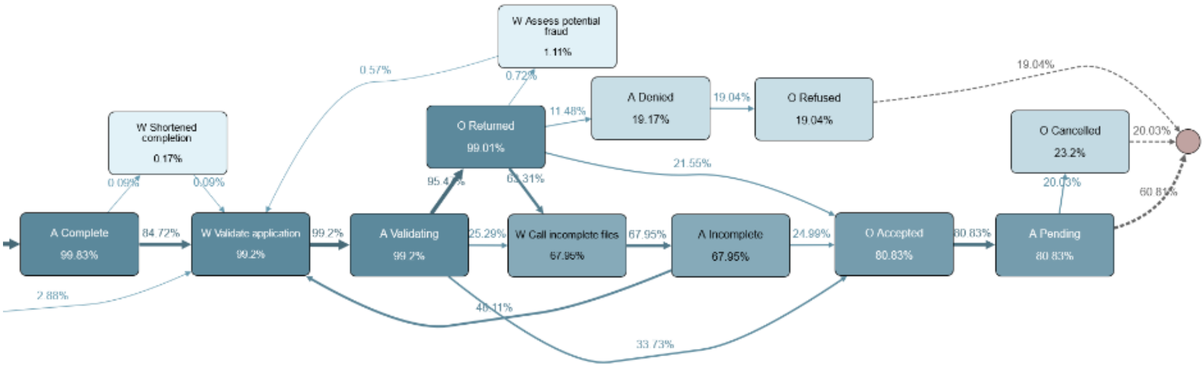
4.4.5 Durations from A_Complete to the end of the case

There are also some significant waiting times in this part of the process(Figure 19. The most noticeable is a long waiting time before the activity A_Cancelled in cases where the application is cancelled. The waiting time is over a month in all pathways except for the longer chain of activities starting with the activity W_Validate_application and even in this chain the total median waiting time is approximately two weeks. It is also interesting that half of the total waiting time in this chain occurs between the activities A_Complete and W_Validate_application which could possibly indicate that the related resources are in some cases overloaded.

The longest waiting times in cases where the loan application was not cancelled also occur just before the activity W_Validate_application further indicating that the related resources may be overloaded. Other factors to point out are the relatively long duration of the activity W_Assess_potential_fraud (which may be explained by the complexity of that activity), the waiting times after the activity O_Returned, and after the activity A_Incomplete.

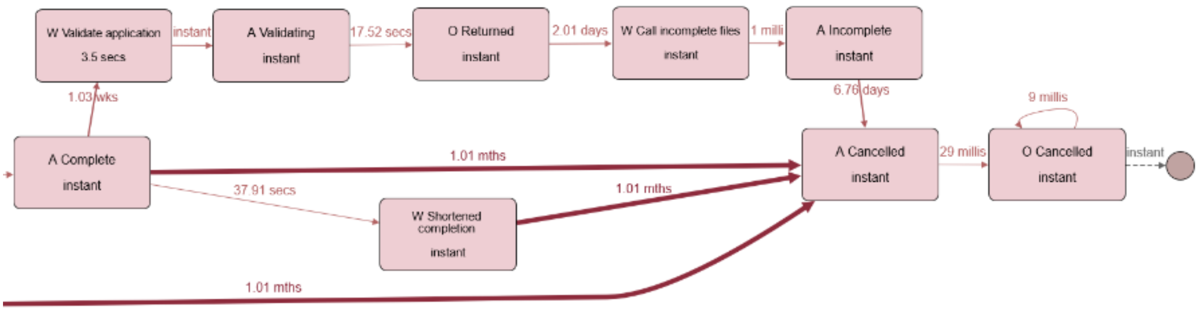


(a) Cases containing A_Cancelled

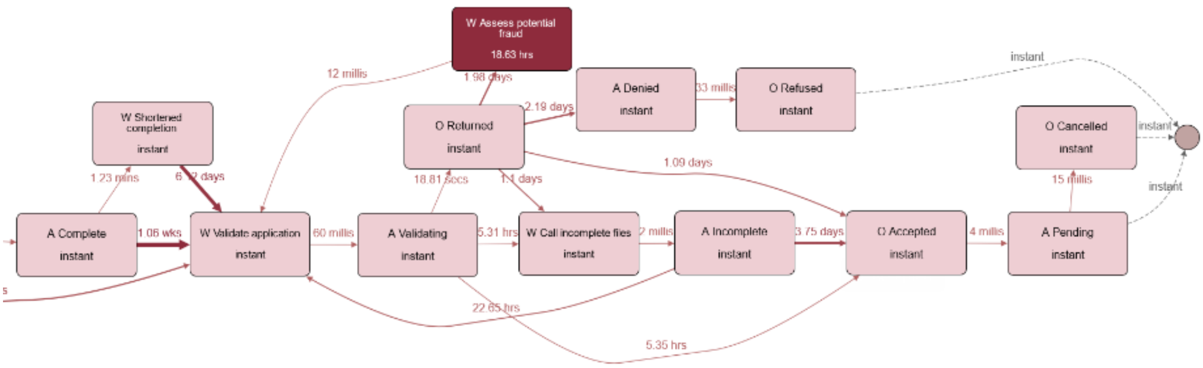


(b) Cases not containing A_Cancelled

Figure 18: Process map from A_Complete to the end of the case with relative case frequency overlay



(a) Cases containing A_Cancelled



(b) Cases not containing A_Cancelled

Figure 19: Process map from A_Complete to the end of the case with the median duration overlay

4.4.6 Summary and Recommendations

The process is relatively similar up until the activity A_Complete in both sets of cases. The only clear difference is that W_Assess_potential_fraud occurs earlier (and takes less time) in cases where the loan application is eventually cancelled. However, it is difficult to offer a concrete suggestion in relation to this observation.

In some outlier cases there are noticeable waiting times of multiple days or even weeks after the activities A_Accepted and O_Created. This may have a negative effect on the conversion as the clients receive the loan offers later than usual. Therefore, it is recommended to set up a system for detecting handling such cases in a timely manner. The most noticeable difference is the long delay before A_Cancelled that occurs in cases where the application is cancelled. This indicates a long inactivity period after which the application is (possibly automatically) cancelled. This adds further support to the recommendation of setting up an (automatic) notification system for the clients.

Some specific resources may be overloaded as there are relatively long waiting times before the activities W_Validate_application, W_Assess_potential_fraud, and after O_Returned A_Incomplete. It may be necessary to review the related procedures and to allocate more resources.

4.5 Task 5: Impact of Application Incompleteness

Guiding question: *Does the frequency of incompleteness influence the final outcome? The bank's hypothesis is that if applicants are confronted with more requests for completion, they are more likely not to accept the final loan offer.*

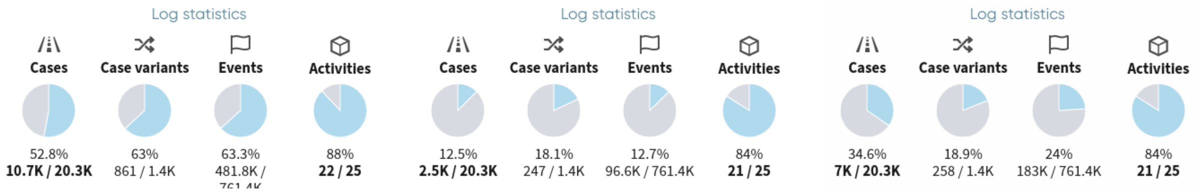
Key results: The number of calls for fulfillment does not affect the outcome of the loan application. If the client is going to accept the offer they will do so regardless of the number of requests for completion. This is likely to not hold for a very high number of calls, however this tipping point was not observed in the given event log.

4.5.1 Analysis

Process outcomes: To analyze the impact of application incompleteness on the outcome of the process, we first need to define the possible outcomes of the process. Given that each process instance relates to exactly one loan application (Section 3) we define process outcomes based on the possible final states of the loan application:

1. A_Pending: All documents have been received, checked, and verified successfully. An offer with a client has been signed with a 53% occurrence rate (see Figure 20a).
2. A_Denied: Bank denies the application with 12.5% occurrence rate (see Figure 20c).
3. A_Cancelled: Client declined the credit offer, didn't't send the documents, or is out of reach for 30 days with 34.5% occurrence rate (see Figure 20b).

Impact of incompleteness: To analyze the impact of application incompleteness on the outcomes we defined above, we begin by filtering the log based on the W_Call_Incomplete_files activity, which occurs in 47.1% of the cases (Figure 21), and we find that the maximum occurrence for that activity in a case is 7. Table 4 shows the percentage of cases based on the number



(a) A_Pending (b) A_Denied (c) A_Cancelled

Figure 20: Statistics of cases filtered based on the outcome of the case

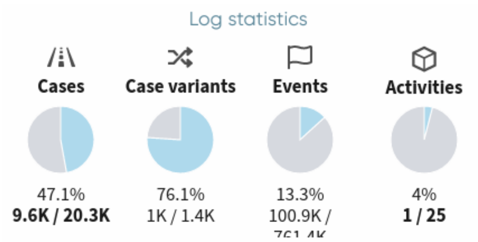


Figure 21: Cases with W_Call_incomplete_files event.

of occurrences of the activity W_Call_Incomplete_files. Accordingly, we split the cases into four filtered event logs: W_Call_Incomplete_files occurs 1, 2, 3, and ≥ 4 times.

Figure 22 shows the performance dashboard for the filtered logs. We can see most of the cases contain W_Call_incomplete_files only once or twice and the graphs for both events and active cases over time are very similar. The average cycle time increases with the the number of occurrences of W_Call_Incomplete_files, however this is to be expected as there is more back and forth between the bank and the client.

Based on Figure 23, we can notice that the frequency of the activities A_Pending, A_Denied, A_Cancelled is similar in all four filtered logs. The only exception is a slightly higher frequency of A_Denied in cases where W_Call_incomplete_files occurs twice, however this activity refers to denial by the bank and not by the client.

4.5.2 Summary and Recommendations

The activity W_Call_Incomplete_files occurs in 47.1% of the cases overall. It occurs once in 30.3%, twice in 12.5% and three times in 3.4% cases. Higher number of occurrences are much more infrequent with the maximum being seven.

Table 4: W_Call_incomplete_files frequencies

Occurrence of "W_Call_incomplete_files"	Number of cases	Percentage of cases
1	30.3%	6155
2	12.5%	2545
3	3.4%	682
4	0.8%	166
5	0.2%	33
6	0%	8
7	0%	2

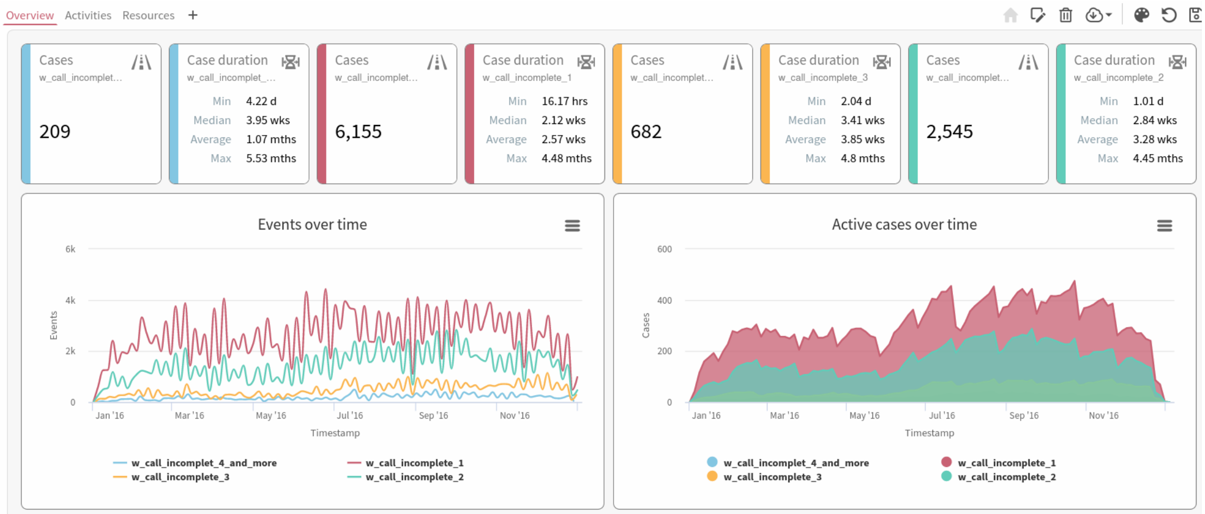


Figure 22: Performance dashboard split by number of W_Call_incomplete_files occurrences

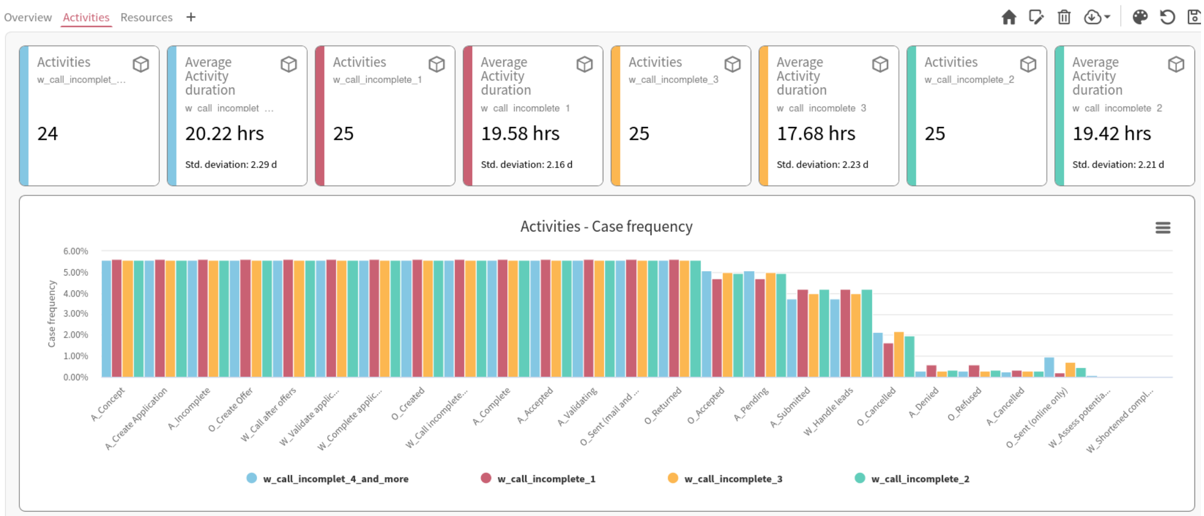
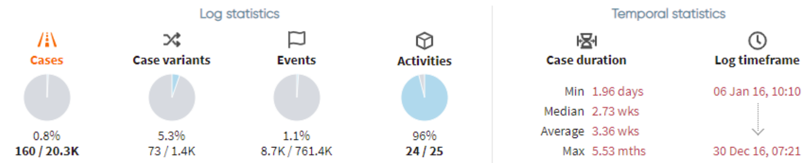
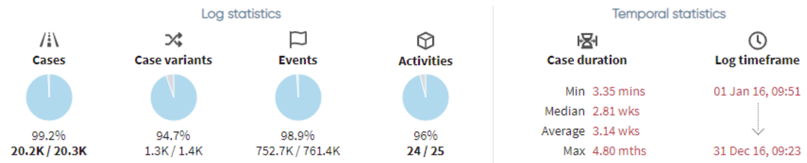


Figure 23: Frequency of activities split by number of W_Call_incomplete_files occurrences



(a) Temporal statistics of the cases with W_Assess_potential_fraud



(b) Temporal statistics of the cases without W_Assess_potential_fraud

Figure 24: Fraud Investigation Analysis

The number of calls for fulfillment does not affect the outcome of the loan application. If the client is going to accept the offer they will do so regardless of the number of requests for completion. This is likely to not hold for a very high number of calls, however such a tipping point was not observed in the given event log. It is difficult to give a specific recommendation, however requests for completion should not be avoided during processing of the loan application.

4.6 Task 6: Impact of Fraud Assessments

Guiding question: *Is there a difference in cycle time between the cases that include a fraud investigation and those that do not?*

Key results: There are only 160 cases that include fraud investigation. The average cycle time of these cases is 1.54 days longer than the cases without fraud investigation. Fraud investigation itself is a relatively expensive activity with an average duration of 3.24 days and a waiting time of 2.6 days before and 1.15 days after the fraud investigation.

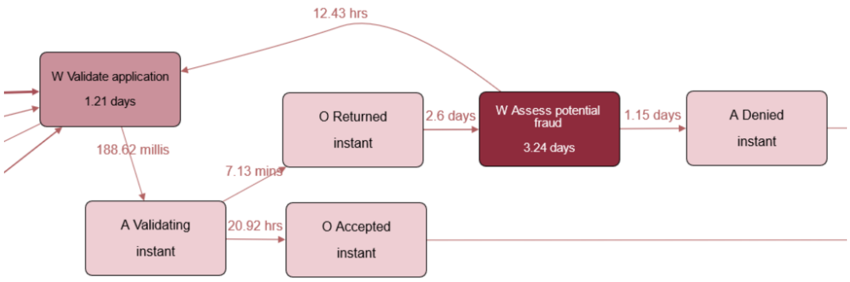
4.6.1 Cycle time

The event log contains a workflow activity called W_Assess_potential_fraud which means that the case has been through an estimation of potential fraud. There are only 160 cases that go through this activity (0.8% of all cases).

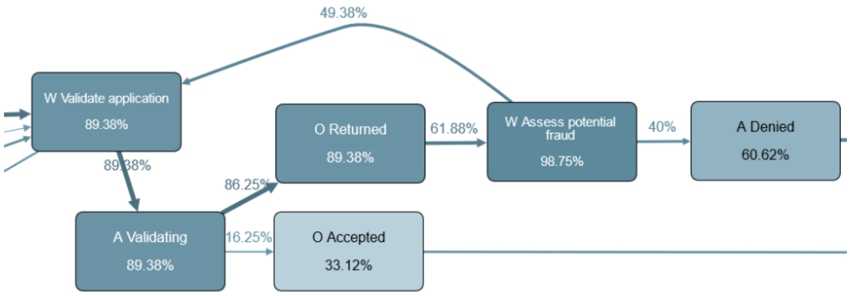
The temporal statistics of these cases are presented in Figure 24a. The temporal statistics of the cases without potential fraud estimation are presented in Figure 24a. On average, cases that go through fraud estimation take 0.22 weeks (1.54 days) longer than the cases that do not go through fraud estimation. Therefore there is a clear difference in cycle times between the cases that include a fraud investigation and those that do not.

4.6.2 Process maps

The process maps related to the activity W_Assess_potential_fraud are shown in in Figure 25. The average duration of W_Assess_potential_fraud is 3.24 days and there is a waiting time of 2.6 days before and 1.15 days after the activity W_Assess_potential_fraud (Figure 25a). This



(a) W_Assess_potential_fraud average duration



(b) W_Assess_potential_fraud relative case frequency

Figure 25: Process maps related to W_Assess_potential_fraud

shows that fraud assessment is a relatively resource intensive activity and that there may be issues with resource availability.

Based on Figure 25b we can see that 40% of loan applications are denied (A_Denied) after fraud assessment and 49.38% of cases continue with the workflow activity W_Validate_application.

4.6.3 Summary and Recommendations

There are only 160 cases that include fraud assessment. Therefore it is a relatively rare activity and does not effect the overall cycle time of the process by a significant amount. However, the average cycle time of cases without fraud assessment is 1.54 days longer than the cases without fraud assessment, so there is a clear difference in cycle times.

Fraud assessment itself is a relatively expensive activity with an average duration of 3.24 days and a waiting time of 2.6 days before and 1.15 days after the fraud assessment. And that 40% of loan applications are denied (A_Denied) after fraud assessment and 49.38% of cases continue with the workflow activity W_Validate_application.

It is likely that foregoing fraud assessment is not possible. However the time effectiveness of this activity should be assessed, especially concerning the relatively long average waiting times before and after the given activity.

4.7 Task 7: Analysis of Multiple Offers

Guiding question: *How many customers ask for more than one offer? Are these offers asked as part of a single conversation or multiple conversations? Does the number of offers matters in terms of conversion rate? In other words, does the number of offers impact on the likelihood of the applicant accepting a loan offer?*

# offers	# cases	Percentage	total # cases
1	15116	74.3 %	**15116**
2	4312	21.2 %	**5227**
3	655	3.2 %	
4	197	1 %	
5	43	0.2 %	
6	7	~ 0%	
7	6	~ 0%	
8	6	~ 0%	
9	1	~ 0%	

Figure 26: Distribution of the number of loan offers per loan application

Key results: Two or more offers are created in 5227 (25.7%) cases. In 99% of these cases the offers are asked in multiple conversations. Creating multiple offers increases the conversion rate from 51% to 57% and decreases both the ratio of cancellations and the ratio of denials.

4.7.1 Number of offers per application

To study how many applicants ask for more than one offer, we applied a repetition filter based on the "O_Created" offer activity that denotes the creation of an offer. We found that 74.3% of cases have only one offer, and 25.7% of applications have more than one offer. Figure 26 shows an overview of the number of offers created per loan application.

4.7.2 Number of conversations for multiple offers

We define the number of offers requested in multiple conversations as offers that have a time interval greater than one day between the O_Created and W_Call_After_Offer events. In around 26% of the applications, the client requests more than one offer. And in about 99% of these cases (Figure 27), customers ask for different offers in multiple conversations (more than one day between the O_Created and W_Call_After_Offer events).

4.7.3 Effect of multiple offers on conversion rate

We defined the conversion rate as the number of cases ending in the A_Pending state divided by the total number of applications. We observed that the conversion rate improved as the number of offers grew. Applications having only one offer have a conversion rate of 51% while applications having two or more loan offers have a conversion rate of 57% (Figure 28 and Figure 29). Additionally, the percentage of both denials (A_Denied) and cancellations (A_Cancelled) is lower for cases having two or more loan offers. Therefore, it seems that it is advantageous to make multiple offers.

4.7.4 Summary and Recommendations

Two or more offers are created in 5227 (25.7%) cases and in a vast majority (99%) of these cases the offers are asked in multiple conversations (i.e. offers having a time interval greater than

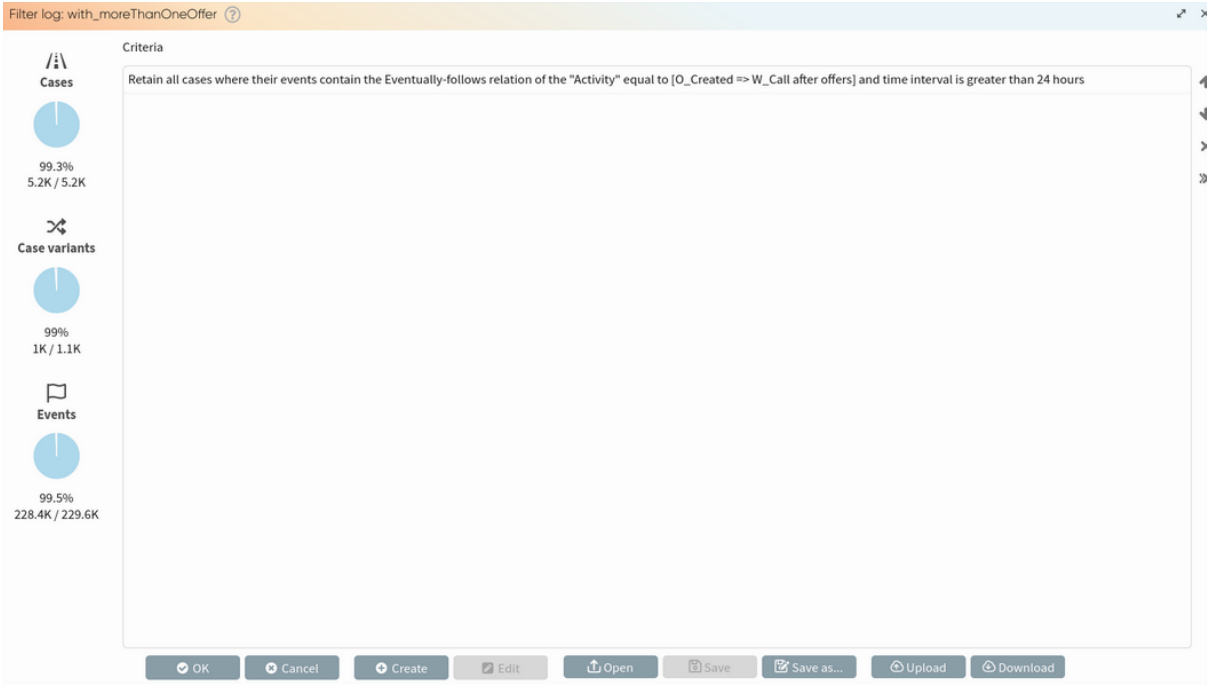


Figure 27: Detecting separate conversations for asking loan offers (statistics on the left)

# Offers	A_Denied	A_Cancelled	A_Pending	Total # Cases
1	2001 (13%)	5356 (35.5%)	7759 (51%)	15116 (74.3%)
>=2	548 (11%)	1689 (32%)	2990 (57%)	5227 (25.7 %)

Figure 28: Impact of the number of offers on the final outcome

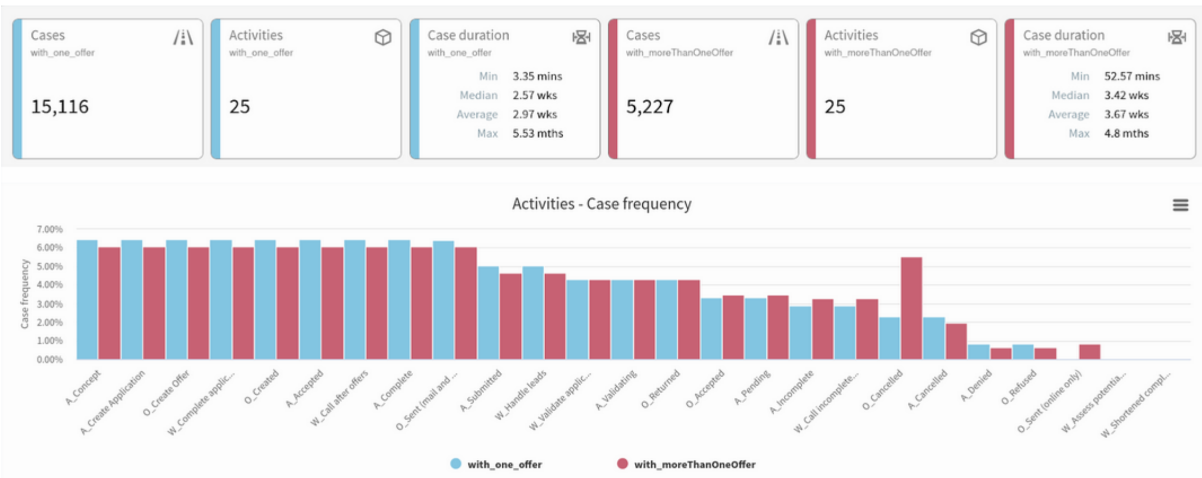


Figure 29: Dashboard comparing the effect of number of offers on activity frequency

one day between the O_Created and W_Call_After_Offer events). It is very rare that multiple offers are asked for during the same conversation.

Creating multiple offers increases the conversion rate from 51% to 57%. Furthermore, it reduces the rate of denials from 13% to 11% and the rate of cancellations from 35.5% to 32%. Therefore, we can conclude that it is advantageous to make multiple offers. And given that most of the multiple offers are not made during the same conversation it is possible that a continuous engagement over time from the bank also has a positive effect.

4.8 Task 8: WIP Analysis

Guiding question: *Is the WIP of the process constant over the timeframe of the log. If not, what factors can explain the change(s) in WIP?*

Key results: The case arrival rate peaks in June, July and August however the number of resources peaks only after August, therefore resulting in a significant backlog of loan applications. This in turn results in a non-constant WIP over the timeframe of the event log.

4.8.1 WIP Analysis

To perform the WIP analysis, we created a custom Python script that is available in the following GitHub repository. The WIP of the process is not constant over the timeframe of the event log. The event log contains events for 364 days of the year 2016. We performed weekly WIP analysis by reporting the average WIP value for every week. From Figure30, we can notice that the average WIP at the beginning of the month is always lower than the average WIP at the end of the month. Figure 31 shows the monthly arrival of cases. The maximum number of cases arrived was in June, July, and August, which explains the peak in the middle of the WIP analysis in Figure 30.

The number of resources from January till end of May was 106. The number of resources during June, July, and August was 114. The number of resources from September till the end of the year was 121. The peak of the case arrival happens in June, July and August, although, during that period, the bank has increased the resources to become 114 working resources, which result in an average case duration of 3.46 weeks. From the comparison of case duration before and after the peak arrival of cases, the increase of resources during the three peak months: June, July, and August, should be the maximum, however, the maximum number of resources happens after the case rush months. Figure 32 shows the difference in case duration summary statistics between the cases during June, July and August, and the cases during the rest of the year.

4.8.2 Summary and Recommendations

Given that the case arrival rate peaks during the summer months it is recommended to add more resources during that time of the year (i.e. June, July, and August). That may reduce the average case duration to 2.98 weeks.

4.9 Task 9: Free-Form Analysis

Guiding question: *Any other interesting insights about factors that affect waiting times, cycle times, or the outcome of the process (the acceptance of a loan offer)? This last part of the project is a*

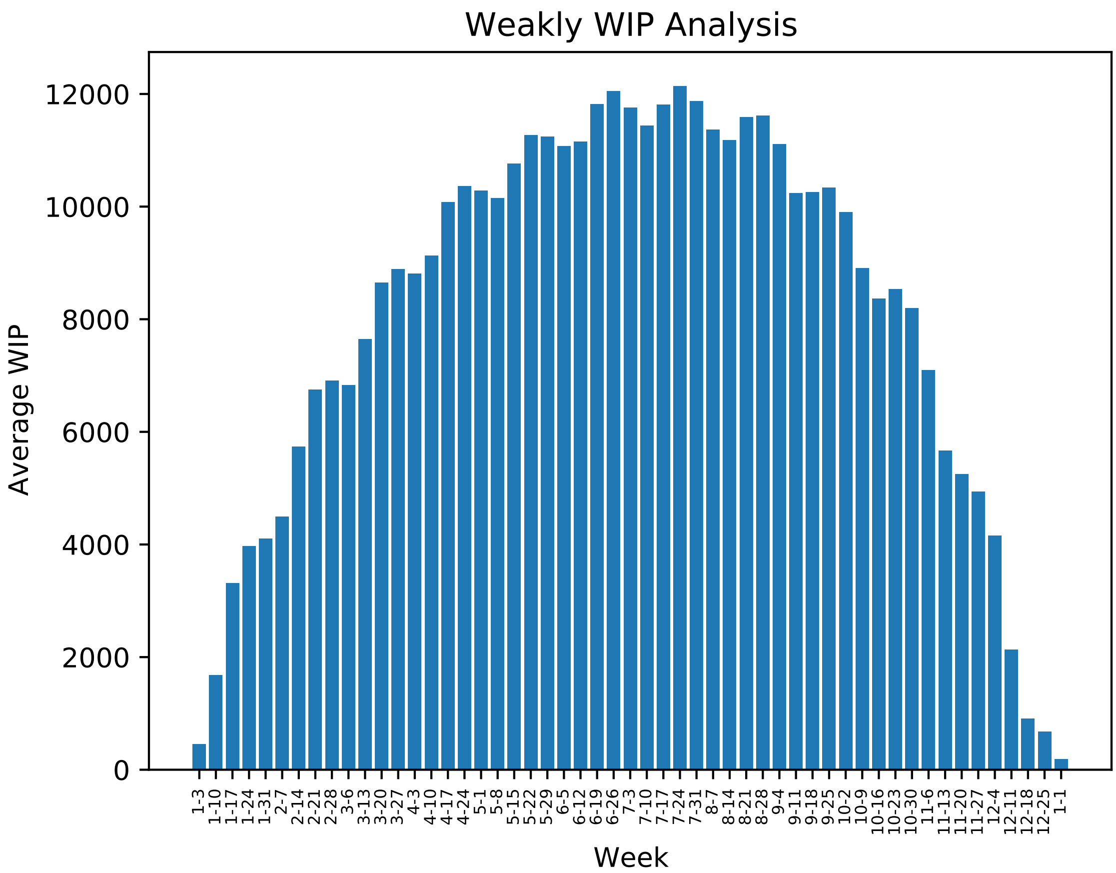


Figure 30: Weekly WIP Analysis

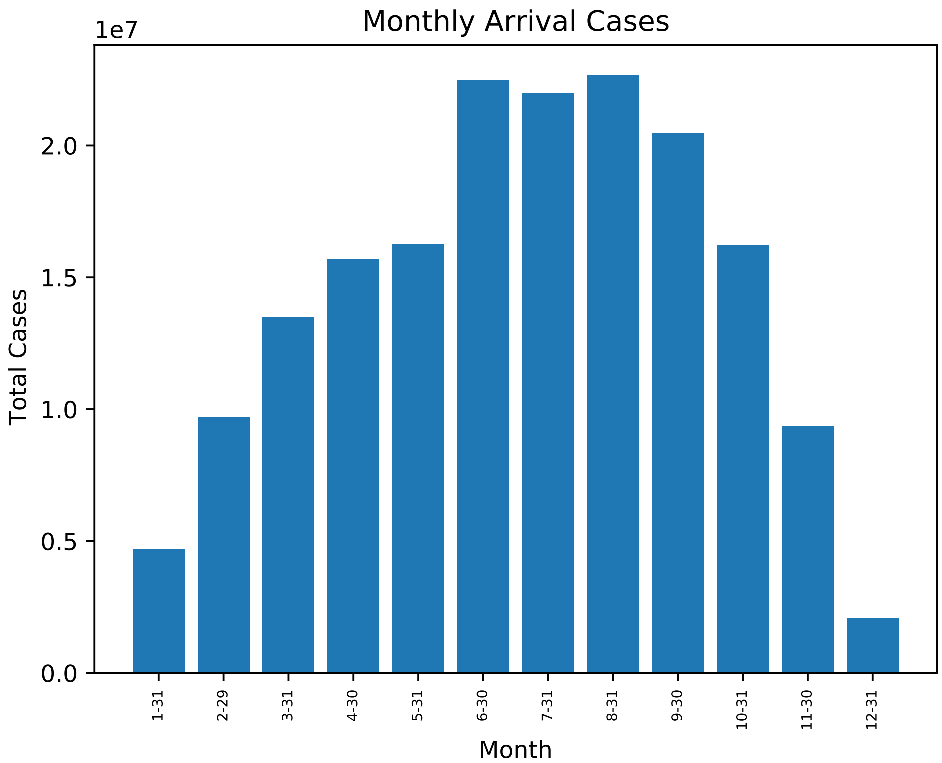
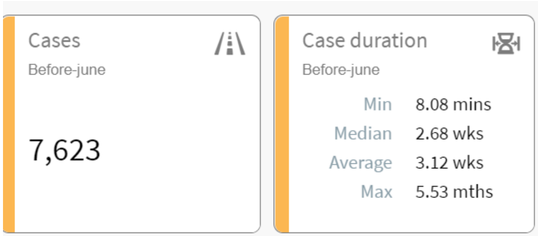
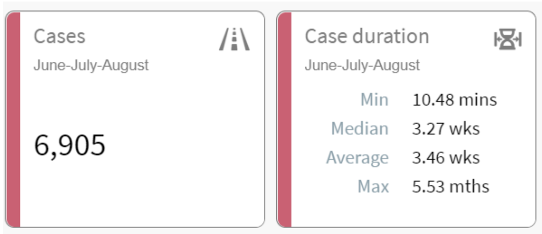


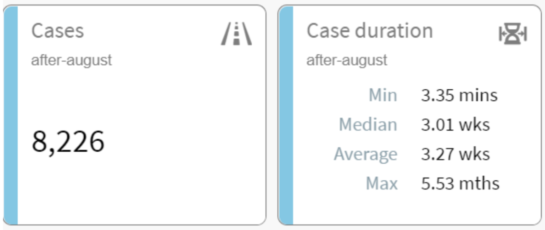
Figure 31: Monthly Case Arrival



(a) Cases from January to May



(b) Cases from June to August



(c) Cases from September to December

Figure 32: Case Duration during June, July, and August VS. during the rest of the year

creative exercise: you decide what do you want to put here. This item is graded subjectively based on the level of creativity of your analysis approach, and how insightful are your findings. For example, you may use machine learning techniques to build a predictive model from the dataset, then use SHAP values to find out how the occurrence of different activities or the occurrence time of different activities affect the cycle time of the process.

Key takeaways: The developed predictive model is able to predict the outcome of an ongoing loan application case with a ROC AUC score of 88.6%. The most relevant features effecting the outcome (based on SHAP analysis) are monthlyCost, numberOfOffers, and creditScore. Based on the results of action rule mining it is possible to optimise the process by for example making more offers to the client under specific circumstances.

4.9.1 Research questions

Based on the free-form nature of the guiding question, we formulated three research questions:

RQ9_1 What will happen in the future for ongoing loan applications? Is the ongoing case going to end with the desired outcome (application the A_Pending state) or not?

RQ9_2 How are the process outcomes predicted? Or which information in the event log effects the predictions the most?

RQ9_3 Can the process be further optimized? What actions we can be taken to increase the probability of the desired outcome?

In the following sections, we will answer the research questions in a detailed manner. To answer **RQ9_1**, and **RQ9_2** we will create a predictive model (Section 4.9.3) to predict the probability of the undesired outcomes with the help of machine learning and SHAP analysis(Section 4.9.4). Additionally, to answer **RQ9_3**, we will build a rule mining technique based on the actionable attributes (Section 4.9.5). The code developed for answering the free-form question is written in Python and can be found in the following GitHub repository.

Table 5: Confusion matrix of the prediction outcomes per bucket.

Output	Number of Buckets
True Deviant	52703
False Regular	12505
False Deviant	11493
True Regular	39290

4.9.2 Data Reprocessing

Before tackling the research questions defined in the previous section we need to prepare the event log to achieve the highest possible prediction accuracy.

Firstly, we found that some of the information in the event log was not in a suitable format for machine learning techniques e.g., number of offers, number of terms, etc. Accordingly, we computed this data based on the information provided in the original log and based on the "O_create offer" activity. We dropped the "offerID" column since it will not add more information to the predictive model. Then the event log was further modified to tackle possible null values, missing values, and irregular cases.

Next, we defined the possible outcomes of a case based on the end state of the loan application, i.e., "A_pending," "A_Cancelled," and "A_Denied". We defined the cases that end with "A_pending" as regular cases with labels equal to 0 and the cases that end with "A_cancelled" or "A_denied" as deviant cases with labels equal to 1. This allows us to build a binary classification model for predicting the probability of undesired outcomes.

4.9.3 Predictive Model

After preprocessing the event log, we have to extract suitable features from the log. We follow the work proposed by Taineema et al[1] to perform the extraction of the prefixes of the traces registered in the log. We performed cluster bucketing and index encoding. The selected machine learning model is XGBoost.

For a better training of the model, we performed parameter optimization for the XGBoost parameters. The source code is available on GitHub repository. The file optimize_params.py perfroms the parameter optimization. The output of the optimization step can be found in the file “optimal_params_xgboost_DahnaLoanApplication_prepared_cluster_index.pickle”. To train the model, first we iterate over each bucked resulted from the cluster bucketing. We perform index encoding. We then train the classifier for the bucket using XGBoost. We evaluate our model using the ROC AUC score, the resulted AUC is: 88.66%. Table 5 presents the confusion matrix of the prediction outcomes. We report the number of buckets for the true/false Regular/Deviant.

4.9.4 SHAP Analysis

To understand the behavior of our outcome-oriented predictive models, we used consolidated frameworks (i.e. ELI5, LIME, and SHAP) to explain how predictions are made using the predictive model. *ELI5* is a Python package that helps to debug machine learning classifiers and explain their predictions in an easy-to-understand and intuitive way. *LIME* is a unique approach created by Riberio Marco, Singh Sameer, Guestrin Carlos to obtain the execution of any base

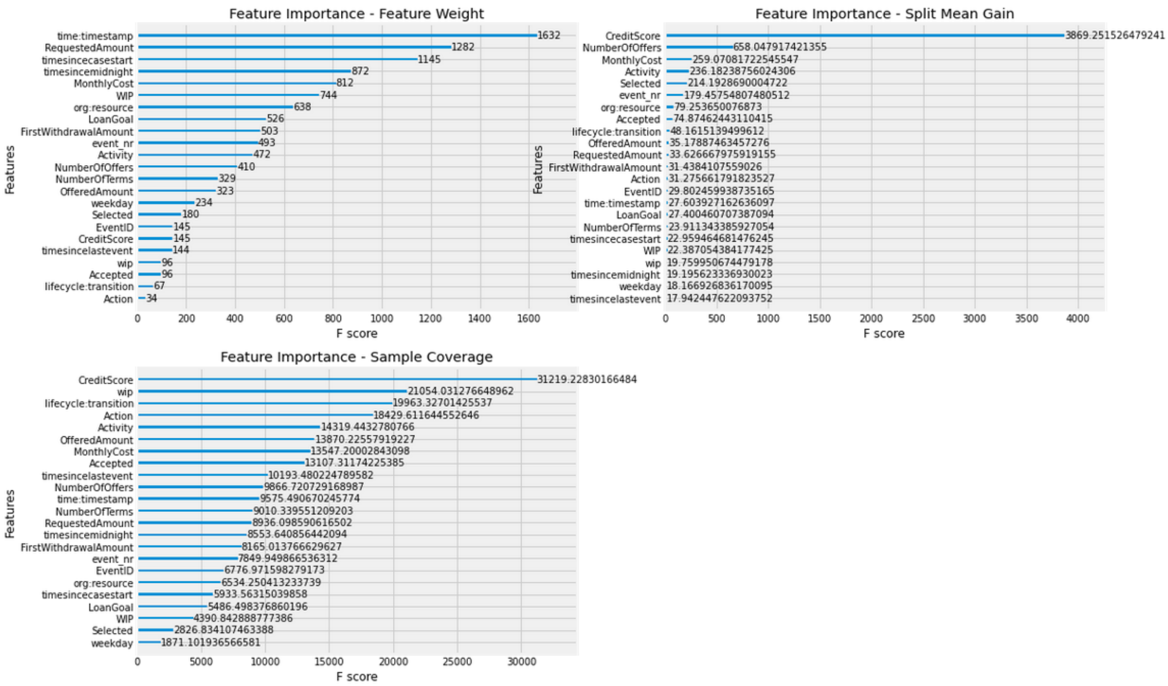


Figure 33: Default Feature Importances from XGBoost

predictive model utilizing interpretable surrogate models (e.g., linear classifier/regressor). Such a form of comprehensive evaluation helps in generating explanations that are locally faithful but may not align with global behavior. *SHAP* Shapley Additive exPlanations gives every feature a weight that refers to the importance of this feature for a specific prediction. Particularly, SHAP values attempt to describe the outcome of a model as a total amount of the effects of every feature being entered into a conditional expectation.

In this part, we started first by trying out the global feature importance calculations that come with XGBoost based on feature weights, gain, and coverage, as shown in Figure 33. We can recognize that the most relevant features are MonthlyCost, nummberOfOffers, creditScore, and WIP.

The next step is to examine ELI5. One of the standard-worthy methods to describe model forecast choices to either a technical or a more business-oriented person is to study each sample prediction. ELI5 achieves that by giving weights for every feature describing the importance of that feature and how it contributes to the final prediction choice over all trees. If we take only one sample as shown in Figure 34, we can notice the common prominent features being the CreditScore, LoanGoal, wip, MonthlyCost, and the NumberOfOffers. Additionally, we explored the results of using LIME since it helps with explaining predictions on tabular (i.e. matrix) data, as shown in Figure 35. We clearly can see that we have the same results as we got from the previous analysis.

Finally, we used SHAP values, which attempt to describe the outcome of a model as a total amount of the effects of every feature being entered into a conditional expectation. In this project, we worked with binary classification problems, which means our target label has two possible values, 0 or 1 (or 0, 1). SHAP provides us with a detailed logic about which features have the highest effect or impact on the model. Figure 36 shows an explanation of features with every feature contributing to shifting the model outcome from the goal value

Actual Label: 0
Predicted Label: 0
y (score **-1.502**) top features

Contribution?	Feature	Value
+0.646	CreditScore	0.000
+0.280	event_nr	4.000
+0.148	LoanGoal	2.000
+0.078	Selected	2.000
+0.015	RequestedAmount	10000.000
+0.013	lifecycle:transition	3.000
+0.013	timesincemidnight	829.000
+0.006	OfferedAmount	0.000
+0.001	wip	3442.000
-0.001	FirstWithdrawalAmount	0.000
-0.008	NumberOfTerms	0.000
-0.024	timesincecasestart	0.000
-0.028	timesincelastevent	0.000
-0.032	weekday	2.000
-0.039	NumberOfOffers	2.000
-0.039	time:timestamp	15375.000
-0.059	Accepted	2.000
-0.061	Action	0.000
-0.114	WIP	3442.000
-0.306	org:resource	16.000
-0.550	<BIAS>	1.000
-0.562	MonthlyCost	0.000
-0.881	Activity	24.000

Actual Label: 0
Predicted Label: 0
y (score **-11.006**) top features

Contribution?	Feature	Value
+1.291	Selected	1.000
+1.233	MonthlyCost	127.930
+0.051	timesincelastevent	5465.083
+0.037	Activity	19.000
+0.034	lifecycle:transition	2.000
+0.029	Action	2.000
+0.029	NumberOfOffers	1.000
+0.007	OfferedAmount	6000.000
+0.007	org:resource	77.000
-0.002	timesincecasestart	5604.283
-0.002	weekday	0.000
-0.010	timesincemidnight	799.000
-0.012	Accepted	1.000
-0.015	event_nr	18.000
-0.030	wip	8868.000
-0.032	FirstWithdrawalAmount	6000.000
-0.045	NumberOfTerms	54.000
-0.049	WIP	8868.000
-0.068	LoanGoal	7.000
-0.127	time:timestamp	98517.000
-0.129	RequestedAmount	6000.000
-0.550	<BIAS>	1.000
-12.654	CreditScore	986.000

Figure 34: Weights of ELI5 for every information describing how important is it

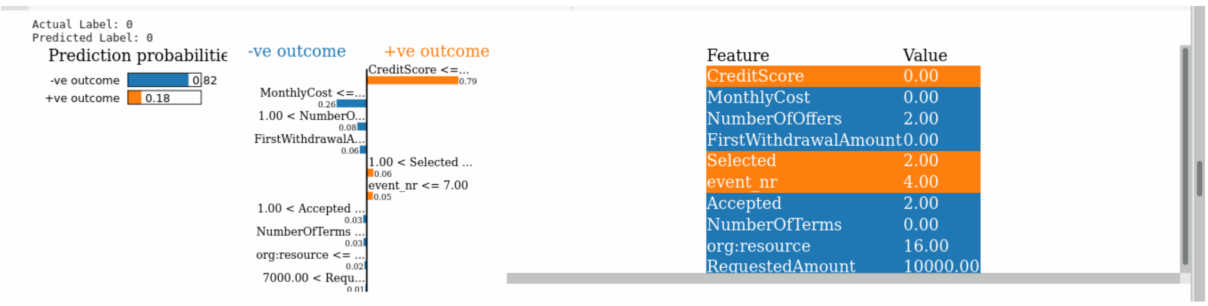
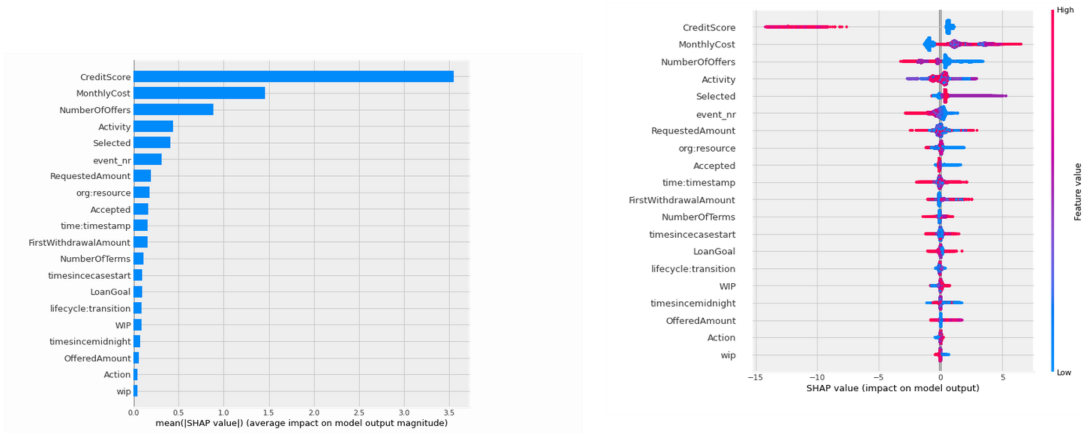


Figure 35: Explaining Model Predictions with LIME

Figure 36: Explaining Model Predictions with SHAP



(a) Feature importance using SHAP

(b) SHAP density plot of SHAP values for every feature to recognize how serious impact every feature holds on the model output.

Figure 37: SHAP feature importance plot and SHAP density plot

(the common model outcome across the training event log we passed) to the original model outcome. Features forcing the forecast higher are displayed in red such as numberOfOffers and MonthlyCosts. The aforementioned figure is related to only one case from the event log, and the outcome for it is negative or false.

Furthermore, Figure 37a shows a traditional bar chart on the basis of the average of the SHAP value measures over the event log and plots it as a simple bar chart. While Figure 37b shows a density scatter plot of SHAP values for every feature to distinguish how much influence every feature has on the model outcome for cases in the validation set. In Figure 37, features are sorted on the basis of the sum of SHAP value measures across overall cases.

After interpreting our outcome-oriented models using ELI5, LIME, and SHAP analysis, we conclude that we have important features that highly affect the final outcome of ongoing cases such as MonthlyCost, numberOfOffers, CreditScore, etc. This analysis leads us to the next section (4.9.5) of the report in which we study the effect of changing values that have an high impact on the outcome of the process.

4.9.5 Rule Mining

In this part, we propose a method that gives us information about what actions or treatments we can apply for ongoing applications to increase the probability of the desired outcome, i.e., the rate of applications ending with the "A_Pending" state. This approach is based on action rule mining [2] that allows us to learn classification rules. The result of action rule mining is a set of action rules which recommend treatments that, if applied, will help to maximize the probability of the desired outcome.

To utilize action rule mining techniques, we first need to define what attributes we can change

```
Stable cols: ['ApplicationType', 'binned_CreditScore', 'LoanGoal', 'binned_RequestedAmount']
Flexible_cols: ['binned_FirstWithdrawalAmount', 'binned_MonthlyCost', 'binned_NoOfTerms', 'MatchedRequest', 'binne
d_NumberOfOffers']

The number of discovered rules are: 25

Rule: r = [(binned_CreditScore: low) ∧ (ApplicationType: new credit) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_N
oOfTerms: 49-60 → 97-120) ] ⇒ [label: 0 → 1] with support: 0.034443958242170405, confidence: 0.4322354134180291 an
d uplift: 0.042314633680989325.

Rule: r = [(binned_CreditScore: low) ∧ (ApplicationType: new credit) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_N
oOfTerms: 6-48 → 97-120) ] ⇒ [label: 0 → 1] with support: 0.034443958242170405, confidence: 0.44852675035791056 an
d uplift: 0.047273331945777375.

Rule: r = [(binned_CreditScore: low) ∧ (ApplicationType: new credit) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_N
oOfTerms: 61-96 → 97-120) ] ⇒ [label: 0 → 1] with support: 0.034443958242170405, confidence: 0.38740050498692397 a
nd uplift: 0.020484549925060894.

Rule: r = [(binned_CreditScore: low) ∧ (ApplicationType: new credit) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_N
oOfTerms: 120+ → 97-120) ] ⇒ [label: 0 → 1] with support: 0.034443958242170405, confidence: 0.42536251505078376 an
d uplift: 0.017814210863181267.

Rule: r = [(binned_CreditScore: low) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_NoOfTerms: 49-60 → 97-120) ] ⇒ [l
abel: 0 → 1] with support: 0.034443958242170405, confidence: 0.4322354134180291 and uplift: 0.042314633680989325.

Rule: r = [(binned_CreditScore: low) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_NoOfTerms: 6-48 → 97-120) ] ⇒ [la
bel: 0 → 1] with support: 0.034443958242170405, confidence: 0.44852675035791056 and uplift: 0.047273331945777375.
```

Figure 38: Extracted rules from the action rule mining approach

that may affect the probability of the outcome. In this report, we showed in section Section 4.7 that the number of offers that the bank creates for each application has a high impact on the probability of the outcome. Additionally, we selected other flexible features that we can change during a running case such as the number of terms, monthly cost, etc based on the analysis we did in section 4.9.4. Accordingly, we prepared the data to fulfill the requirements of the action rule mining approach with the help of the method proposed by [3].

We executed the action rule mining technique on the prepared event log with support = 3 and confidence = 55, finishing in 25 rules with maximum support = 0.13, maximum confidence = 0.44, and maximum uplift = 0.04. Sample of the extracted rules are shown in figure 38, and 39. If we look at the first extracted rule in figure 38, we can see that if the credit score of the application is low and the application type is new credit, then it is beneficial to two offers (instead of one) and to double the number of terms from 60 to 120. Under the given conditions these two changes have support of 3% and uplift of 4% to change the final outcome of the case from an undesired outcome to a desired one.

4.9.6 Summary and Recommendations

The developed predictive model is able to predict the outcome of an ongoing loan application case with a ROC AUC score of 88.6%. It would be beneficial to develop an on-line monitoring solution for tracking ongoing loan applications and to apply the predictive model on the monitoring results. This would allow early detection of loan applications that are likely to end with an undesired outcome, which in turn would allow to take further measures for effecting the outcome.

Based on the results of action rule mining it is possible to optimise the process by for example making more offers to the client under specific circumstances. The most relevant features effecting the outcome (based on SHAP analysis) are monthlyCost, numberOfOffers, and creditScore and this is also reflected in the discovered action rules. Most of the modifications are technically not complicated (effecting the number of terms for example) and therefore the related changes can be made in a timely and cost-effective manner.

```
Rule: r = [(binned_CreditScore: low) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_NoOfTerms: 61-96 → 97-120) ] ⇒ [l
abel: 0 → 1] with support: 0.034443958242170405, confidence: 0.38740050498692397 and uplift: 0.020484549925060894.

Rule: r = [(binned_CreditScore: low) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_NoOfTerms: 120+ → 97-120) ] ⇒ [la
bel: 0 → 1] with support: 0.034443958242170405, confidence: 0.42536251505078376 and uplift: 0.017814210863181267.

Rule: r = [(ApplicationType: new credit) ∧ (binned_FirstWithdrawalAmount: 7500-9895 → 9896-75000)  ∧ (binned_Numbe
rOfOffers: 1 → 2) ] ⇒ [label: 0 → 1] with support: 0.04057010689504282, confidence: 0.3541938292360742 and uplift:
0.049084082226685116.

Rule: r = [(ApplicationType: new credit) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_NoOfTerms: 49-60 → 97-120) ]
⇒ [label: 0 → 1] with support: 0.040382571732199786, confidence: 0.3423819382597532 and uplift: 0.0299048006985180
72.

Rule: r = [(ApplicationType: new credit) ∧ (binned_FirstWithdrawalAmount: 7500-9895 → 0-7499)  ∧ (binned_NoOfTerm
s: 49-60 → 97-120) ] ⇒ [label: 0 → 1] with support: 0.04975932987435144, confidence: 0.3530418024296795 and uplif
t: 0.016279023282174614.

Rule: r = [(ApplicationType: new credit) ∧ (binned_NoOfTerms: 6-48 → 97-120) ] ⇒ [label: 0 → 1] with support: 0.13
408764143276863, confidence: 0.3172988148341573 and uplift: 0.029559121932800636.

Rule: r = [(ApplicationType: new credit) ∧ (binned_NumberOfOffers: 1 → 2)  ∧ (binned_NoOfTerms: 6-48 → 97-120) ] ⇒
[label: 0 → 1] with support: 0.040382571732199786, confidence: 0.3636719604749456 and uplift: 0.03611564265073894.

Rule: r = [(ApplicationType: new credit) ∧ (binned_FirstWithdrawalAmount: 7500-9895 → 0-7499)  ∧ (binned_NoOfTerm
s: 6-48 → 97-120) ] ⇒ [label: 0 → 1] with support: 0.04938425954866538, confidence: 0.38499996995427044 and uplif
t: 0.018914509894436278.
```

Figure 39: Extracted rules from the action rule mining approach

5 Conclusion

In this report we presented the analysis of the loan application handling process in an European bank Dhana. The analysis is in part guided by a combination of specific questions proposed by Dhana and by the process mining expertise of the authors of this report. The analysis is based on an event log of all loan applications filed with Dhana in 2016.

Overall the process has a relatively clear structure, indicating that the process has been specifically designed at some point in time. There are clear entities (loan applications containing possibly multiple loan offers) that the process is managing and clear life cycle of process instances is visible trough the statuses of the managed entities.

However, as also suspected by Dhana, there are some issues with the given process which result in a non-optimal performance. The SLA requirements of different loan types are violated with an high frequency and it is likely that no loan type based prioritisation is employed. There are some activities that stand out in terms of long processing times and long waiting times (e.g. Application validation and Fraud assessment, but also cancellation of a loan offer). Some of the aforementioned processing times and waiting are likely a result of inefficient or insufficient resource allocation.

Furthermore, based on a combination of the questions proposed by Dhana and the insights gained from the analysis of the event log, it can be claimed that some of the assumptions made by Dhana do not hold at this point in time. Some of the proposed questions indicate an underlying assumption that the clients should be bothered as little as possible. However, the analysis of the given event log strongly indicates that a more active engagement with the clients would lead to a higher overall conversion rate. Possible solutions would be to implement an (automatic) notification system for the clients, creating multiple offers per loan application and/or providing more guidance for the clients in general.

The report summarises the key findings and recommendations for each of the analysis tasks as well as providing a detailed overview of the results. Additionally, the last section of the report provides a model for predicting the outcome of an ongoing case and multiple fine-grained optimisation suggestions based on prescriptive modeling. This, together with the other sections of the report, provides a full description of how to improve the given process in the future.

References

[1] I. Teinemaa, M. Dumas, M. L. Rosa, and F. M. Maggi, "Outcome-oriented predictive process monitoring: Review and benchmark," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 2, pp. 1–57, 2019.

[2] L. Sỳkora and T. Kliegr, "Action rules: Counterfactual explanations in python," 2020.

[3] Z. D. Bozorgi, I. Teinemaa, M. Dumas, M. La Rosa, and A. Polyvyanyy, "Process mining meets causal machine learning: Discovering causal rules from event logs," in *2020 2nd International Conference on Process Mining (ICPM)*, pp. 129–136, IEEE, 2020.