BPI2017 Challenge: Business process mining–A Loan process application

Introduction- Roei Michael , Avichay Hayne and Asaf Koren

Process mining, the practice of extracting knowledge from event logs recorded by systems, plays a pivotal role in comprehending and optimizing complex operational processes across various fields, including banking. This project, in particular, focuses on the application of process mining to the BPI2017 Challenge dataset. This dataset, made available by the BPI Challenge 2017, comprises various application traces within a banking environment. The project aims to glean insights, identify patterns, and propose enhancements that could potentially revolutionize the operational workflows of financial institutions.

The BPI2017 dataset originates from the system logs of a financial institution. Through a meticulous examination of this data, we aim to map/discover all process flows and investigate any possible inefficiencies. By focusing on the frequency of events, we anticipate identifying points of improvements, thus, enhancing the overall efficiency and effectiveness of banking operations. The project also entails searching for behavior patterns that might enable the institution to perform more in-depth analysis, suggesting changes, improvements, corrections, and learning from its processes.

In the contemporary economic environment, financial institutes face stiff competition, especially from emerging Financial Technology (FinTech) firms. One strategy to withstand this competition is by enhancing customer experience through the application of digitalization and automation techniques that streamline loan processes. Consequently, this project's goal is not only to analyze the loan application process, but also to focus on improving the customer experience and potentially increase revenue.

By addressing these aspects, our project seeks to answer critical questions raised by the process owners. These include queries related to throughput times for various parts of the process, the influence of the frequency of incompleteness on the final outcome, the number of customers requesting more than one offer, and the comparison of conversion between applicants who receive a single offer and those who receive multiple offers.

By intertwining process mining with the banking sector, we hope to unlock profound insights that could help financial institutions improve their operational processes, customer experience, and ultimately, their bottom line. Through this project, we wish to demonstrate the untapped potential and wide applicability of process mining in contemporary financial practices.

Preprocessing

The preprocessing phase aimed to prepare the dataset for a smooth and effective process mining activity. It was performed in a stepwise manner, systematically reducing the size and complexity of the dataset, while ensuring to maintain and highlight the most critical and informative parts of the process.

Trace Frequency Filter:

Our first step was to filter out all traces that occurred only once in the log file. The assumption behind this action was that infrequent traces, specifically those that occurred only a single time, were less likely to be representative of the common process pathways in our application system and hence, could be removed without significant loss of information.

This can be expressed as follows:

Let T be a set of all traces in the log file and n(t) be the number of occurrences of a specific trace t in T. Then, the reduced set of traces T' after this preprocessing step is given by:

$$T' = \{t \, in \, T : n(t) > 1\}$$

Event Redundancy Filter:

We noticed through a visual inspection of the event log that some events seemed to contribute little to no new information about the processes. These included duplicated events and others that were not essential to the process. We removed such events, simplifying the event logs while still maintaining the overall process information.

Creation of Offer Dataset:

In order to enrich our understanding of the processes and enhance our analysis, we created a separate dataset focused on the offers part of the applications. This included important information such as the loan goal, requested amount, offered amount, and credit score. This dataset, while being valuable on its own, also served as an additional dimension to our primary event log data, allowing us to derive deeper insights about the process.

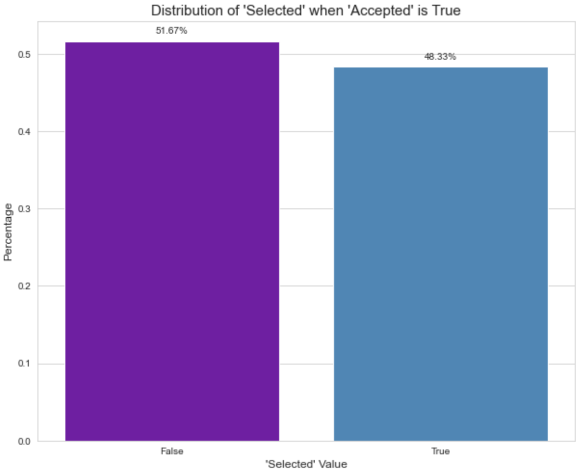
Additional Filtration Methods:

Finally, we devised and implemented several other filtration methods that were more oriented towards improving the visual analysis and exploration of the dataset. These included:

- **Top X Trace Filter**: This filter reduced the dataset to include only the top X most frequent traces, thereby focusing on the most common process paths.
- **Application Outcome Filter**: This filter selectively included only those traces that ended in either 'O_accepted' or 'O_rejected', i.e., the processes which resulted in a definitive outcome for the application.
- **Workflow Segment Filter**: This filter included only the events that occurred between 'W_complete' and 'A_accepted', thereby focusing on the main workflow of the application process.
- **Outliers filtration**: by using the Offers dataset we created we looked through the information about the offers created in the bank and filtered out outliers by examining columns like amount of offers, monthly payment, number of terms, etc.

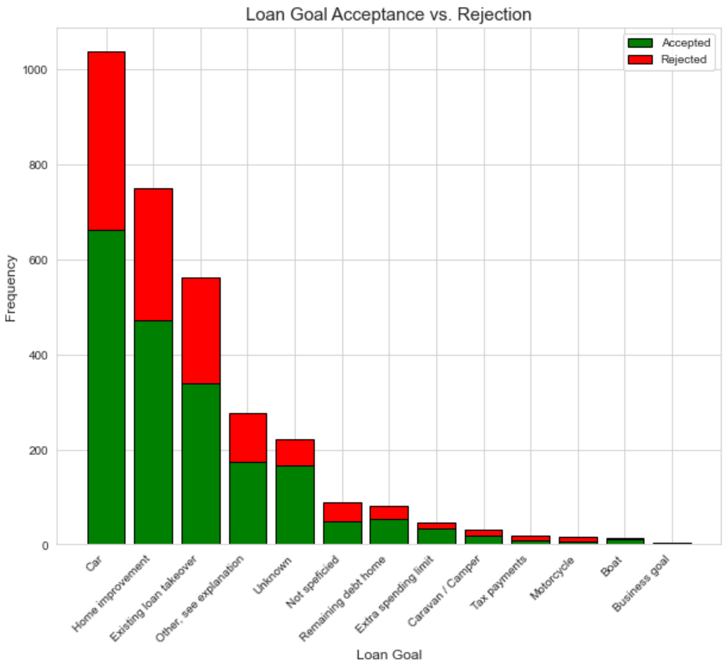
In addition to the filtration we conducted on the data set we crafted a few graphs that helped us understand the data better:

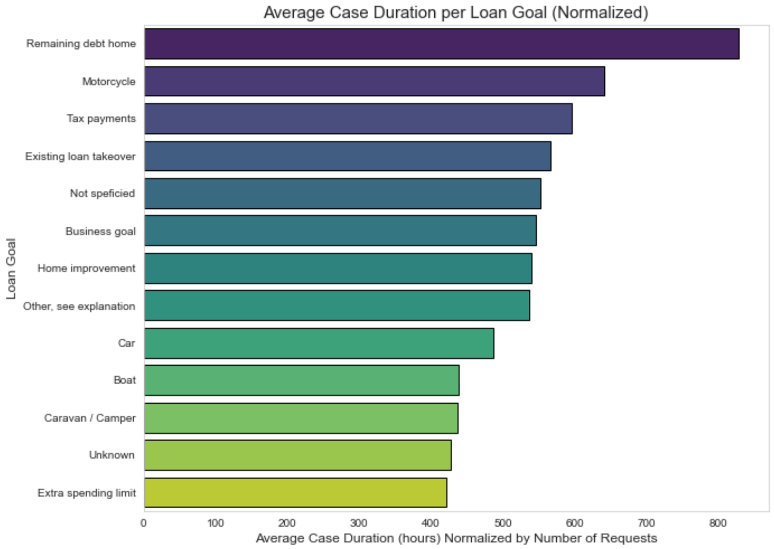
the first analysis we had to do was showing the amount of accepted vs rejected offers :



From it we saw that we have almost a balanced dataset with a slight skewness to rejected offers.

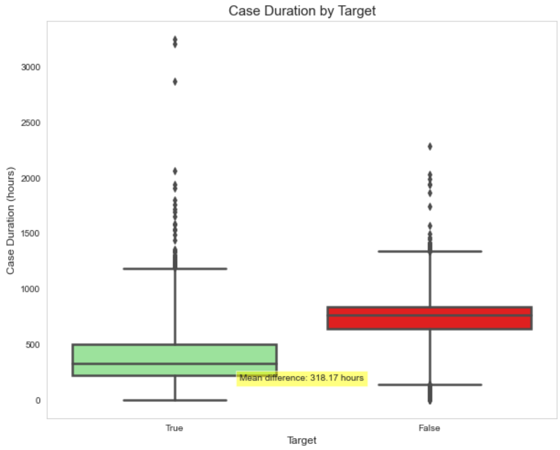
the next analysis we performed was based on the Loan Goal for each application , we measured the time and result of the application and shown them in a form of bar charts:

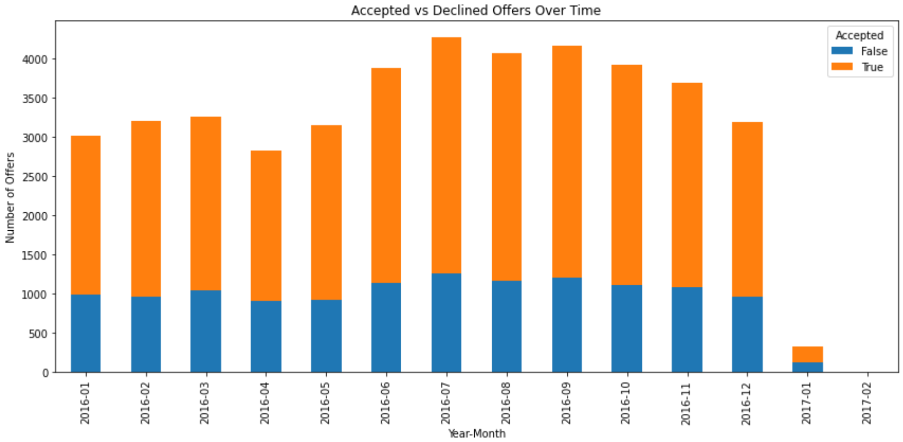




We noticed that most of the applications are related to cars and home while the time they take isn't as less common loan goals, considering that we made an assumption that goals like remaining house debt might lead to a long trace with multiple events that is dragged across a lot of time and could cause bottlenecks.

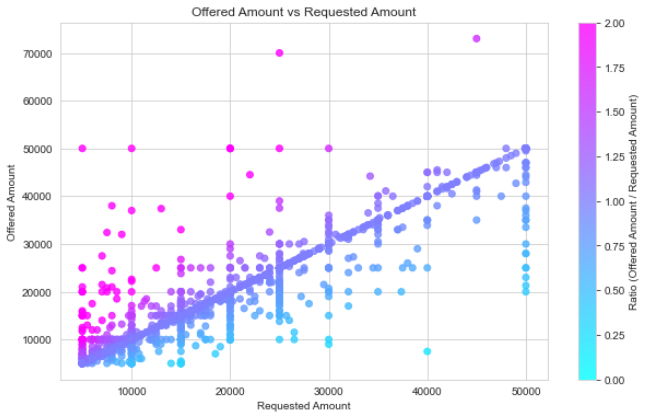
we than looked at each case duration and compared them in regard to the way the offer ended.

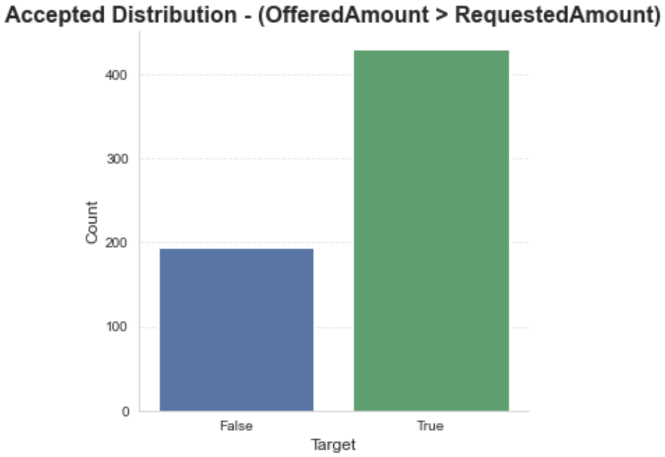


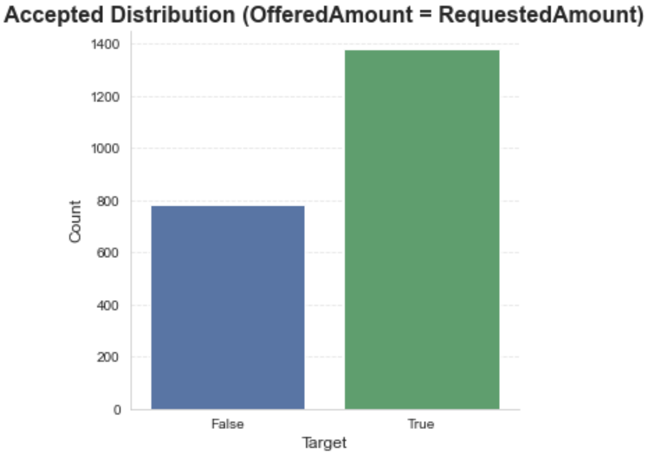


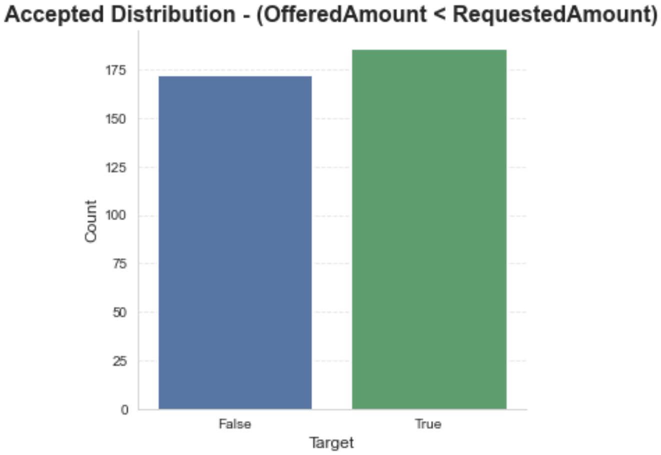
We saw from the graphs that across the months there wasn't a real difference in rejected offers but there were a bit more accepting of offers in the summer months. Another thing to notice is the big gap in average time for case duration comparing between accepted and rejected offers, it did make sense that rejected applications probably got dragged out to multiple offers and we could consider applications with more than a single offer to be more time consuming for the bank and focus our analysis a bit more in that section.

We than looked at the requested to offered amount in each offer to the bank to make predictions on how an offer is going to go based on the ratio of requested to offered amount:





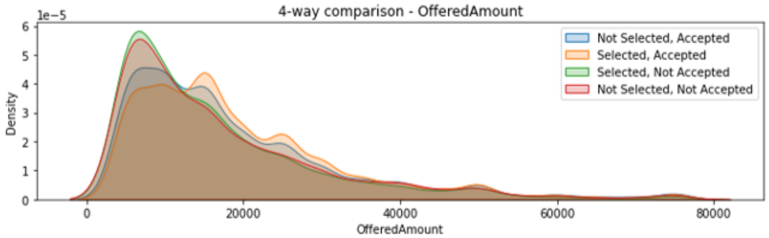


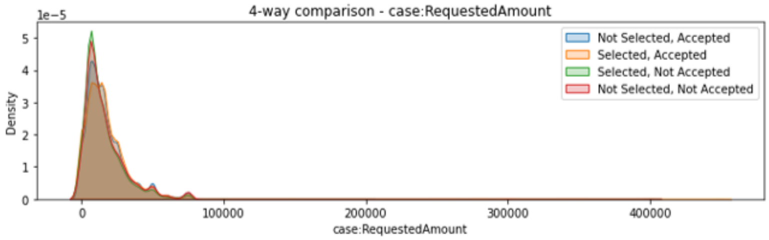


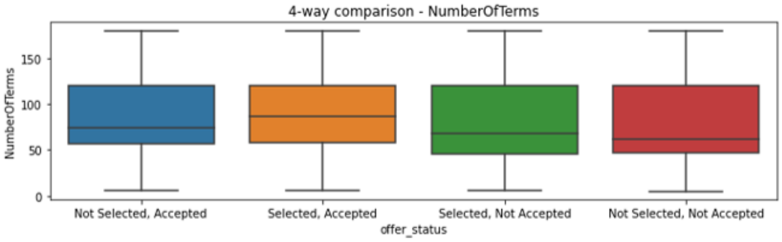
What we saw from these graphs is that most offers are met with the same amount of offer to requested basically giving us the linear purple line in the first graph, than we have 3 zones to analyze , when the offered amount is equal to the requested amount which we saw that ends up in accepted offers most of the time, and the next to zones are when the client and the bank are at a disagreement either the bank offers less or more to the client which led us to seeing how it affects the result of the application. Where when the offered amount is less than requested we see a rise in the percentage of rejected offers drastically.

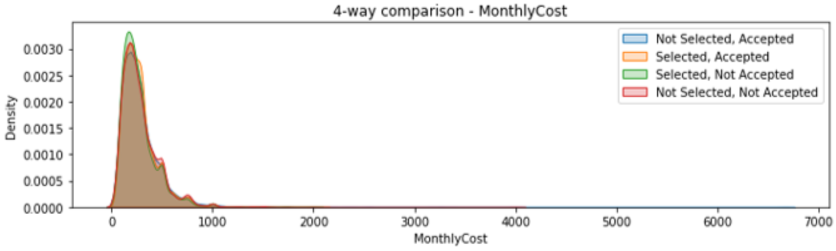
The last analyzation we performed was separate our data to 4 “classes”:
(Accepted , Selected); (Accepted , Not Selected); (Not Accepted , Selected);
(Not Accepted, Not Selected) **Accepted- by the bank , Selected- by the client

And than comparing the offered amount , requested amount , number of months for payment and amount of money needed to be paid each month as follows:









We noticed that there wasn’t much of a change in the 4 classes and it seems that there isn’t a quick determination that could be done to early classify if the offer will be selected/accepted based on the early information like the offered and requested amounts.

Analysis Results

Our analysis unearthed notable correlations and repeating patterns within the process, hinting towards potential areas for process optimization.

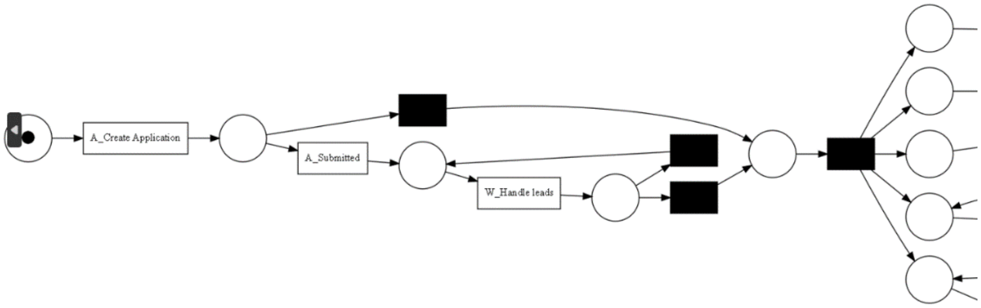
We used the filters mentioned in our preprocessing phase, in order to better handle and inspect various aspects of the dataset and the process. We divided our analysis into 3 main parts: Graphical representation and analysis of various aspects of the process using a representing data set, process mining discovery of selected datasets which describe different parts of the loan application process, and throughput analysis of frequently visited activities which we suspected to be bottlenecks.

Process petri-net building and analysis:

Part I:

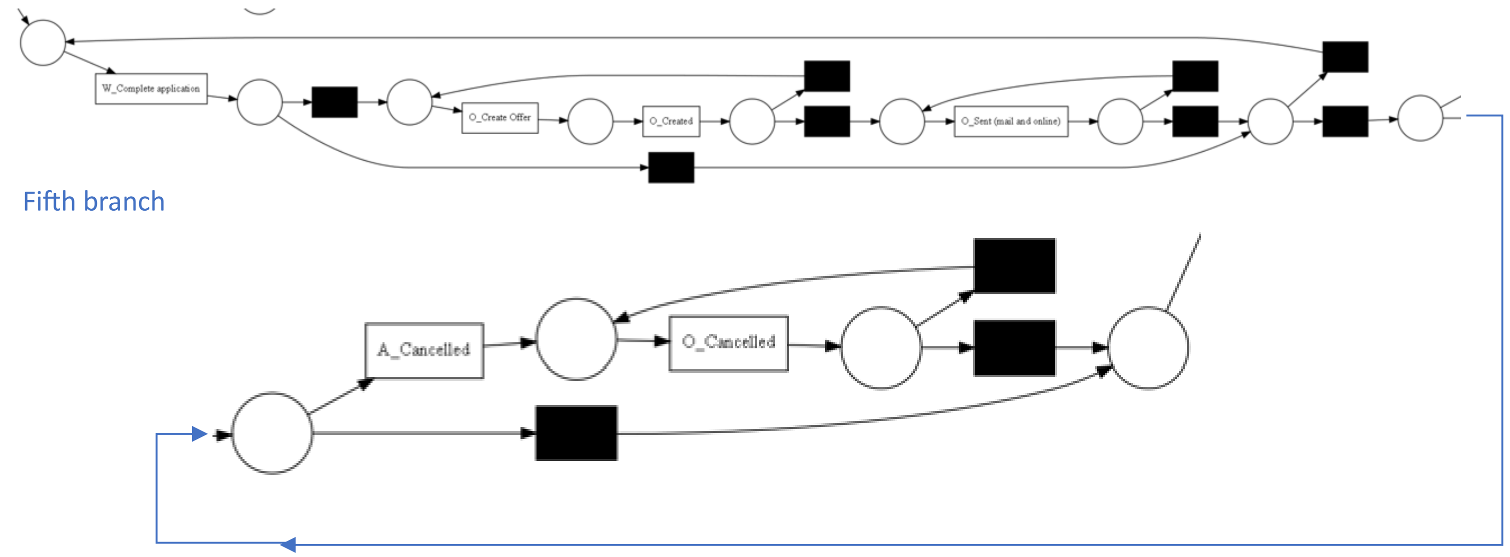
In order to capture the essence of the loan process inside the organization we need to ask ourselves what the bread and butter of loans for the bank are? To answer that question, we took the top 50 variants of traces and built a petri net using the inductive miner, in order to understand the process to the fullest we have decided to explain the petri nets in sections:

The first -

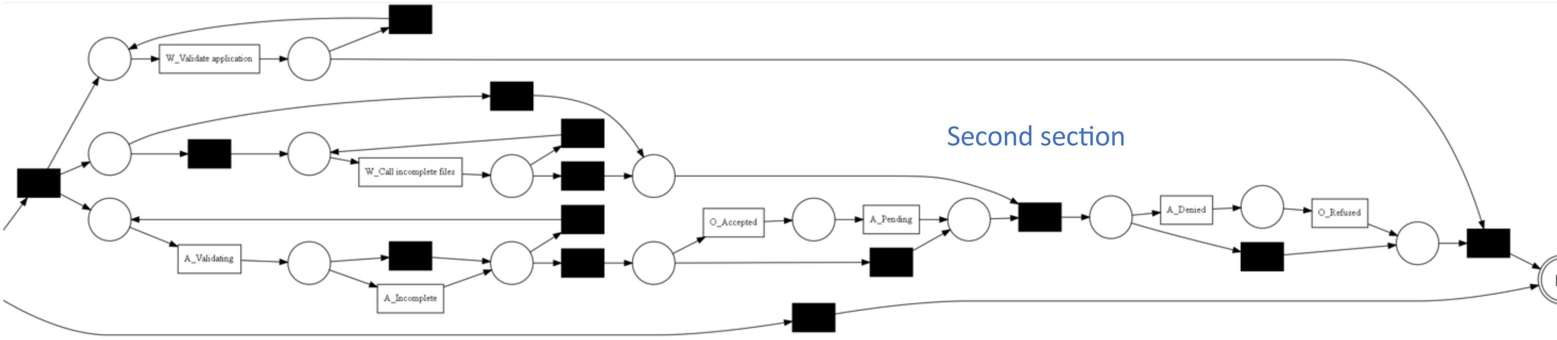


in this part we can see the application being processed inside the bank, the difference between the upper and the lower branches is in the way of submitting the application, for the upper branch the application was submitted via the bank itself, so a banker saw the application and submitted it himself. As for the lower branch the application was submitted online and there was a need for a banker to handle it. After the submission we can see that there are 5 conditions that has to be met in order to continue:

the first condition is for the bank to accept the application, the second is for the bank to create an initial concept for the loan. The third is the offer has been sent to the customer and the bank waits for the customer to send a signed offer and the rest of the required documentation. The fourth, A bank employee contacts the customer to follow-up on the offer. He/she gathers the thoughts of the customer about the offer. If the customer indicates he/she is not interested in the offer or does not answer the inquiries, both the offer and the application will be cancelled. And the fifth, for the actual offer to be created and or canceled:



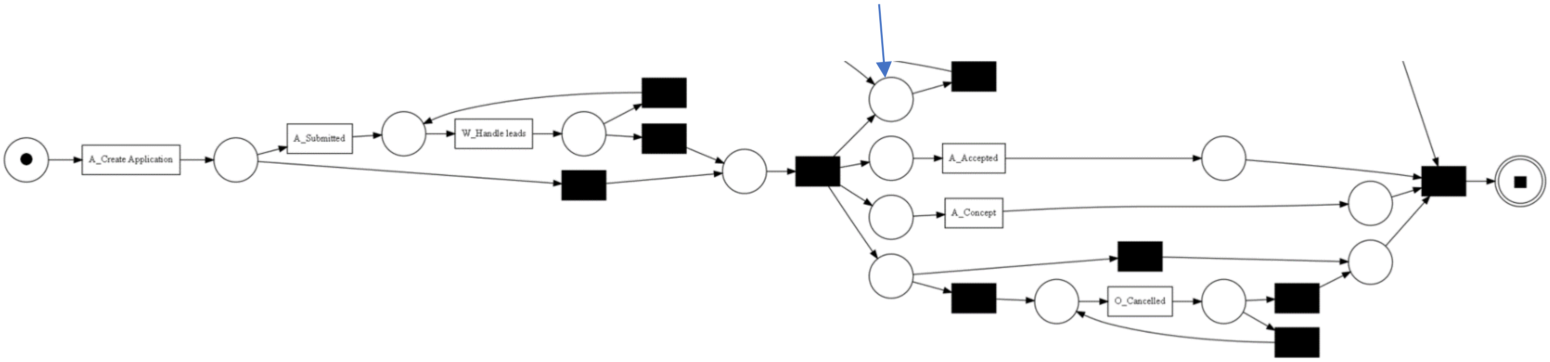
After those 5 conditions had been met there is now the second section of the process, if the application was canceled, we jump to the output instantly, otherwise we have a lot more to the process.
there are again 3 criteria that need to be met for the process to finish, the first is to validate the application with the bank, the second is to call about incomplete files, the third is to validate the files, the second and third branches are combined into a single branch that decides the fate of the application. Whether it will be accepted of refused.



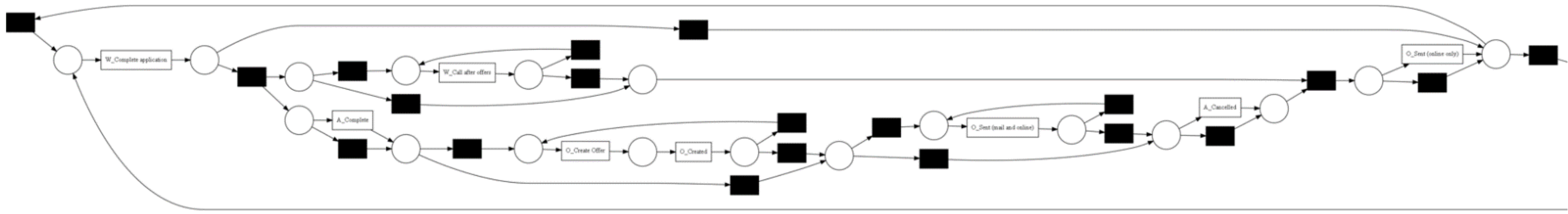
Part II:

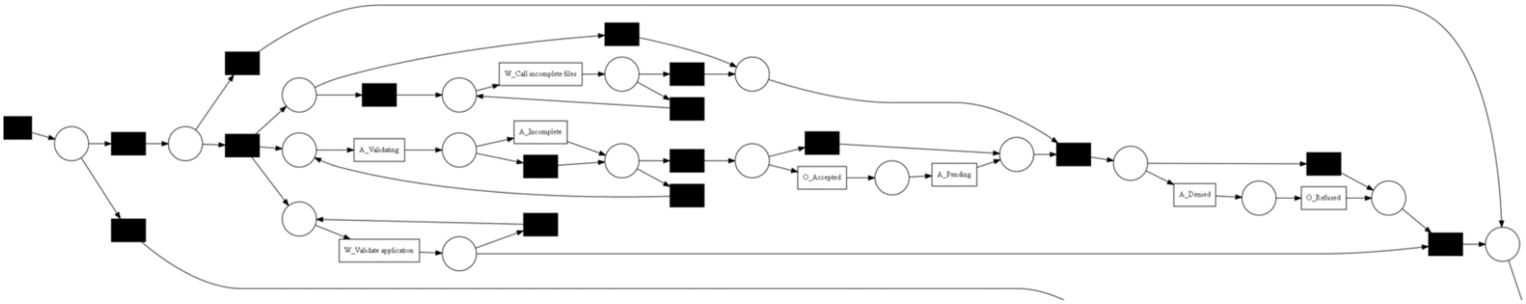
In this section, instead of discussing the essence of the process, its “ideal petri net”

We chose to apply a more forgiving type of filters, first we have created a file that contained the information about each application without writing any additional info like events or type of events, from there we took built a normal distribution graph of the ‘mothlyCost’ and ‘numberOfTerms’ individually and then we took values that helped us take out outliers applications from the event log, finally we took the top 250 variance from the log file and only then we use the inductive miner. In the new Petri net, we can notice a few changes some are significant and some are less. The most notable change It's not in the events that are taking place inside the process but it's in the way that they are organized inside the Petri net. For example, in part one, we had two sections to our Petri net whereas in this Petri net It's more like a one long process rather than two separate sections we can see that instead of the five initial conditions from petri net 1, there are 3 primary conditions and one more branch if the application get canceled.



by the second section of the top50 variants with minor changes: more events are linear to each other rather than being parallel to one another. There are for example instead of calling to the client in parallel to finishing the application we now first finish the application and only then start communicating with him via phone calls.





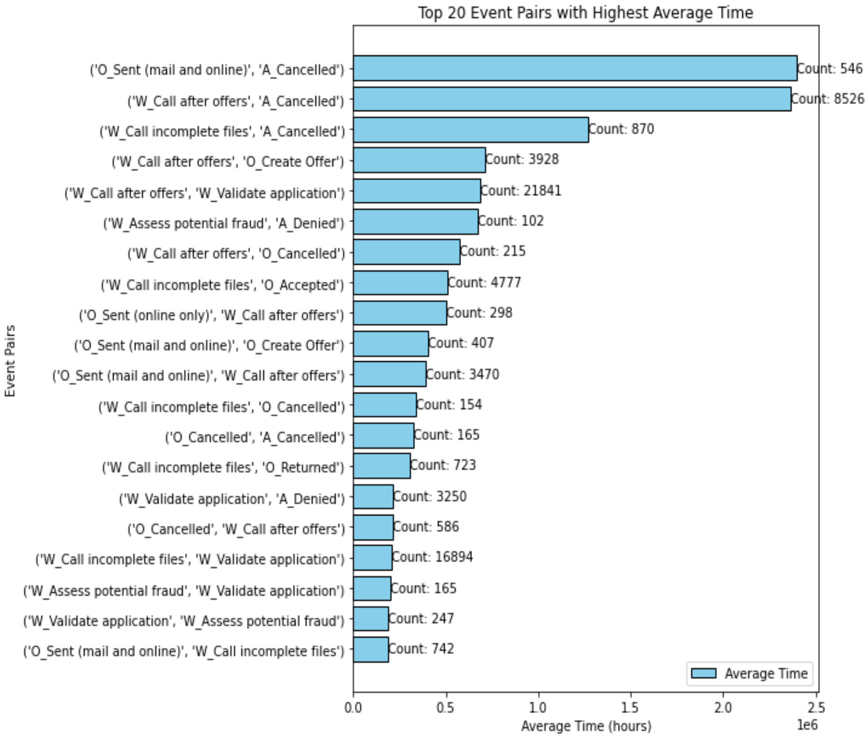
Average trace fitness: 0.99775474.

Log fitness: 0.997390.

Percentage of fitting traces: 86.134.

Data regarding event pairs with highest Average Time

In this part, we conducted an analysis on the event log data to identify bottle necks, patterns and derive insights that could enhance the banking process. We performed several steps of filtering and ordering on the data set, in order to calculate average time differences between event pairs which stood out in regard to average time and occurrence.



Next, we calculated the time differences between consecutive events within each trace. These time differences served as a crucial metric for measuring the duration between events and identifying potential bottlenecks or patterns. By examining these time differences, we aimed to gain valuable insights into the sequential relationships and durations between different events. After carefully examining the information regarding average time, number of occurrences of pair of events and examining the logistical limitations on the pairs we have arrived at several insights from our analysis.

We should note that there are a few more pairs which their average time is much more significant, but the next reasons are why we chose not to address them as bottlenecks:

1) there is a large number of events between them.

2) chronologically the second event is happening before the first event, but, the offer was rejected (a conclusion which take time to reach to) and so the second event is happening again, and so an illusion that a bottle neck occurs is created.

The main and most problematic bottle neck is with the ‘W_callIncompleteFiles’ to ‘O_Accepted’ pair of events. first and foremost the 2 events are happening in parallel and are not directly dependent on one another, the issue lies within the reason those 2 events are happening, every time the A_Validating event deems the application incomplete it continues to the A_Incomplete event, in parallel the bank calls the client about the incomplete application and asks them for the files, this process is repeated until the A_Validating event deems the application as complete only then the application is accepted (O_Accepted), the reason it is so problematic is because this process should not repeat itself so many times. the bank is supposed to be clearer and make sure the applicants understand what is needed from them in order for the whole process to go smoothly. We understand that sometimes the requirements are not understandable by first glance. Our suggestion is for the bank to upgrade their Information Department.

The second bottle neck is (W_callAfterOffers, W_ValidateAppllication).
by observing the petri net we derived from the event log we understand that there are no actual events in between them, only waiting for the offer to be created and sent to the client, we know the described process is not taking a lot of time after analyzing the amount of time all of the events took and not seeing them in our graph. The reason the transition (W_callAfterOffers, W_ValidateAppllication) takes so long is unclear.
overall, we recommend for the bank to launch an investigation to the calls department and understand where the process is lagging behind.

Discussion and Conclusions

The project has provided valuable insights into the bank application process from a process mining perspective. Our iterative approach involving preprocessing of the dataset, exploratory data analysis, and construction of process models in the form of Petri nets has highlighted key aspects of the process, including frequent pathways, time durations, and decision outcomes.

The preprocessing stage was instrumental in refining our dataset, thereby allowing us to focus on the most relevant traces and events. Our methods, which included the trace frequency filter, event redundancy filter, and several others like the Top X Trace Filter, Application Outcome Filter, and Workflow Segment Filter, paved the way for a more streamlined and informative analysis. The creation of the 'Offers' dataset also proved to be a significant enhancement, enabling us to conduct a more comprehensive investigation and filtration of outliers in our data.

Our subsequent exploration and visualization of the data further refined our understanding of the process. By analyzing several key features, such as the final result of an application, the loan goal, the processing time, and the comparison of offered and requested amounts, we were able to discern patterns and relationships that would otherwise have remained hidden.

Finally, the construction and analysis of Petri nets not only visually represented the process but also facilitated the identification of potential bottlenecks and disparities in event transition times. Through iterative modeling, we demonstrated how the application of different filters simplifies the Petri net model, thereby making it easier to interpret and analyze.

In conclusion, this project exemplifies the power of process mining techniques in uncovering insights from complex process data. It has revealed key areas of inefficiency and bottlenecks in the bank's application process that could be the focus of future improvements. Furthermore, it has showcased how preprocessing and model construction techniques can be iteratively applied to generate understandable and valuable process models. The outcomes of this project validate the application of process mining in real-world business process analysis, and it opens the door for future research in this domain. This project can serve as a benchmark for further explorations in process mining, not only in banking but also in various other sectors that rely heavily on complex processes.

References

List all the resources you used or referred to in your project.

1. Scheithauer, G., Henne, R., Kerciku, A., Waldenmaier, R., & Riedel, U. (n.d.). Suggestions for Improving a Bank's Loan Application Process based on a Process Mining Analysis. Metafinanz Informationssysteme GmbH, Munich, Germany.

2. Blevi, L., Delporte, L., & Robbrecht, J. (2017). Process mining on the loan application process of a Dutch Financial Institute BPI Challenge 2017. KPMG Technology Advisory, Brussels, Belgium. Retrieved from https://home.kpmg.com/be/en/home/insights/2017/09/process-mining.html

3. PM4Py. (n.d.). PM4Py Documentation. Retrieved from https://pm4py.fit.fraunhofer.de/documentation

4. Khademolhossieni, P. (n.d.). BPI Challenge 2017. Prezi presentation. Retrieved from https://prezi.com/8pa_nlaoikol/bpi-challenge-2017/

5. Araújo leite, J., & Tavares Sant'anna, M. (2017). Using Process Mining Techniques to Support Improvement in a Financial Institution Process. Rio De Janeiro, Brazil.

6. Jeong, D., Lim, J., & Bae, Y. (n.d.). BPIC2017: Business process mining – A Loan process application. Department of Industrial and Management Engineering, POSTECH (Pohang University of Science and Technology).