Automated Machine Learning in a Process Mining Context

Stijn Kas, Ruben Post, and Sebastiaan Wiewel

Faculty of Science, Utrecht University, Domplein 29, 3512 JE Utrecht
{s.f.kas,r.m.post,s.o.wiewel}@students.uu.nl

Abstract. Process mining techniques have matured over the past years and are now providing valuable insight to organizations and their processes. Still, the topic is developing and the field is actively inspiring new researchers to participate. With the Business Process Innovation (BPI) challenge 2020, students and professionals are challenged to extract insights from a real-world event log. In this report, business questions of the Eindhoven University of Technology are investigated and cost predictions are made of the declaration process of the TU/e. By utilizing a novel automated machine learning approach, automated model assembly, comparison, and hyperparameter optimization is introduced in the context of process mining. The results show that an AutoML model can predict the amount a case will overspend with an average error of €24.

Keywords: BPI Challenge 2020, Process Mining, Machine Learning, AutoML

1 Introduction

The annual Business Process Innovation (BPI) challenge invites both students and professionals to analyze a real-life event log, by focusing on one or more process owner's questions or other unique insights into the process(es) captured in the event log [7]. This year, the event log originates from the period 2017/2018 and consists of travel declaration made by personnel of the Eindhoven University of Technology (TU/e) [7]. Just like most other organizations, the TU/e staff travels to visit conferences, disseminate knowledge, and meet colleagues in the field. Typically, the company covers the travel expenses, as they do for the TU/e employees.

The goal of this report is to answer the business questions asked by the TU/e using traditional process mining techniques. Answering these questions will pose immediate value for the TU/e and is therefore considered of great importance. Additionally, to provide a more in-depth look into overspent declarations, this report further analyzes overspending throughout the provided event logs. In combination with traditional process mining techniques, data mining algorithms are used to perform three analyses. The following analyses are performed to try to 1) determine whether there is a difference in the behavior of traces that overspent, 2) predict whether the trace will overspend, and 3) predict how much

will be overspent. Knowing whether a trace will overspend will allow the TU/e to put in place preventive measures that will reduce the number of cases that overspent. Knowing how much will be overspent will allow the TU/e to better budget the amount spent on travel costs both for cases that will overspend and for cases that will happen in the future. Lastly, insight into the behavior of traces that overspent might help the TU/e prevent this kind of behavior.

Section two explains the methods that give this report its structure. Some aspects of the methods have been substituted to fit the needs of this project. The third section describes the data. The event logs provided by the challenge are briefly depicted and a selection is made. The fourth section answers the business questions asked by the TU/e. The fifth section prepares the data for more traditional data mining techniques. This data preparation is needed to make the data fit the need of these more traditional techniques. The sixth section analysis this data to create a model that can predict the amount a case will overspend. The last section, section seven, concludes the report and section eight provides a discussion and directions for future work.

2 Method

The project is structured along the Process Diagnostic Method (PDM) described by [1]. This method is chosen because it provides general guidelines on how to approach a process mining project and has easily interchangeable phases. In Figure 1, the phases of the method are shown. In this method, the control flow analysis, performance analysis, and role analysis are inherent phases. However, for this report, these analyses are not used for predictions. Therefore, the log preparation, log inspection, and transfer result phases are kept but the analysis phases are substituted by prediction analysis (please note that this change is not reflected in Figure 1). Below the figure, each phase is described.

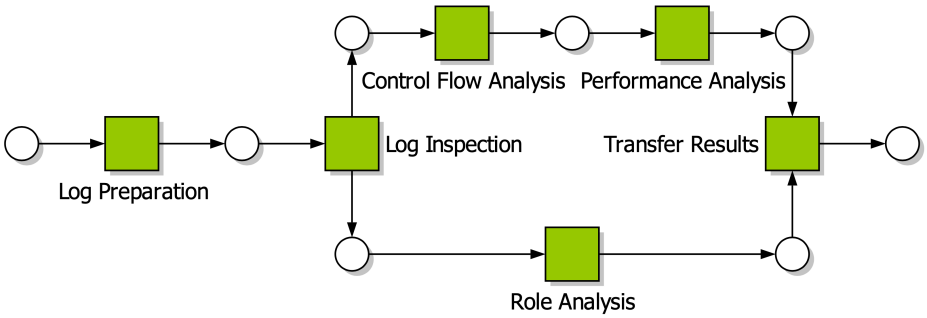

Figure 1: Method phases [1]

1. **Log preparation:** This phase consists of extracting and transforming the data. The first step of this phase is to select the best notion of a case. Next, the activities and events are identified. For this project, the data is already extracted from the information systems of the TU/e. However, the notion of a case is important because it influences the perspective on the data. In this project, the perspective of the data is changed throughout the analysis. However, since this change in perspective is done to facilitate traditional data analysis techniques, the steps taken to change this perspective are described in section three. The other activity performed (extracting the data from the information system) in this phase are already performed by the facilitators of the BPI challenge 2020. hence, no further description of this phase can be given.
2. **Log inspection:** This phase serves for the researchers to get familiar with the data. Here, the data is transformed to fit the need of the desired analysis. In section three, the log is inspected and details about the log are given. Additionally. to fit the need of traditional data analysis, data preparation steps for these analyses are described in section five.
3. **Analysis phases:** Which analyses are performed here is dependent on the goal of the project. [1] describe three types of analysis that provide various perspectives and insights on the data. However, in this project, a different analysis is performed. This is why the control-flow and organizational analysis phases are substituted with a prediction analysis phase. Thus, even though some phases are substituted, the project still adheres to the structure of the method. In section six, the performed analyses are described. In section seven, the results of the analysis phases are described. Additionally, in section four, business questions are analyzed and answered.
4. **Transfer results:** This phase is where the results of the analysis phases are discussed with the process owner. In the method, the authors propose to gain insight into the behavior seen in the system [1]. However, because in this project some analysis phases are substituted, this phase will consist of expanding upon the results of the analysis. In section six, the results are discussed alongside their limitations. Lastly, section eight will conclude the research and provide directions for future work.

In addition to PDM, a complementary element of the PM^2 method is adapted into the research project [3]. Compared to PDM, the PM^2 is a more generic approach to process mining. The method defines 5 phases that center around an iterative analysis of a business question. Similar to the phases of PDM, PM^2 describes extraction and preparation phases before continuing with analysis before finally putting the results into practice. However, PM^2's general and iterative set-up allows for non-sequential execution of the project. The element that is adapted to this research is the iterative approach to the analysis phases of PDM, allowing for more flexibility if the analysis does not provide satisfactory results.

3 Data description

The BPI challenge 2020 provides five data sets in XES format (called "event logs"). The XES standard defines a grammar for capturing systems behavior by means of event logs [4]. This means that the data has also been extracted from the information systems at the TU/e and transformed into this standard. The following event logs are provided:

1. `Request For Payment`: 6,886 cases with 36,796 events executing non travel related payment requests;
2. `Domestic Declarations`: 10,500 cases with 56,437 events executing domestic travel declarations;
3. `Prepaid Travel Cost`: 2,099 cases with 18,246 events executing payments for prepaid travel expenses;
4. `International Declarations`: 6,559 cases with 72,151 events executing international travel declarations;
5. `Travel Permits`: 7,065 cases with 86,581 events executing all related events of relevant prepaid travel costs declarations, travel declarations, and travel permit requests.

Each event log gives a different perspective on the processes performed at the TU/e. However, not all event logs are used in the desired analyses. The first part of the analysis aims to answer a subset of business questions provided by the TU/e regarding International Declarations and Travel Permits. The second part of the analyses looks at the overspent amount in travel declarations and the event logs, therefore, should at least provide the requested amount, the declared amount, and (as verification) the overspent amount. These values can be found in the `International Declarations` and `Travel Permits` event logs. Thus, this report will use these two event logs. The event logs contain several attributes on event level:

```
International Declarations:
id: Declaration ID, Permit ID
resource: The resource who performed the event,
activity: The activity that is performed by the resource,
timestamp: The timestamp of the completion of the task.

PermitLog:
id: id, Declaration ID
resource: The resource who performed the event,
activity: The activity that is performed by the resource,
timestamp: The timestamp of the completion of the task.
```

There are also several attributes on a case level, meaning most cases have values for the following attributes:

```
International Declarations:
role, Permit travel permit number,
```

DeclarationNumber, Amount, RequestedAmount,
Permit Tasknumber, Permit BudgetNumber, OriginalAmount,
Permit ProjectNumer, concept:name,
Permit OrganizationalEntity, travel permit number,
Permit RequestedBudget, id, Permit ID, permit id,
BudgetNumber, Permit ActivityNumber, AdjustedAmount.

Permit Log:
role, OrganizationalEntity, ProjectNumber, TaskNumber,
ActivityNumber, TotalDeclared, concept:name, RequestAmount,
Overspent, travel permit number, DeclarationNumber,
id, RequestedBudget, BudgetNumber, OverspentAmount.

OverspentAmount, AdjustedAmount, TotalDeclared, RequestedAmount, and RRequestedBudget are attributes with continuous values. ProcessDuration is calculated as a continuous attribute as well. To give some insight into the values of these attributes, each of them have been visualized in a box plot, see Appendix A. The box plot shows the mean of the attributes and the first and the third quantile (representing the median of the lower and upper half of the attribute respectively). The other attributes are also explored. Note that these attributes are taken from both the International Declaration and Permit Log event log. process duration is the time that a process takes to complete for the International Declaration log.

Besides the continuous attributes, the process is visualized to get a better understanding of how the process is performed and which decisions are often made. To make the process comprehensible, the process models are based on the most frequent trace variant (a trace is a specific sequence that a case can follow). To get a better representation of this trace variant, we cluster all similarly behaving traces with the ActiTraC plugin in ProM [2]. ActiTraC is a model-based clustering algorithm that provides clusters of similar behaving traces with a fitness of 1.0 (meaning it only allows for behavior seen in the event logs that are contained in the cluster). This algorithm is used to provide a more comprehensive overview of the entire event log in one visualization. For the International Declaration event log, this algorithm provided a cluster containing 53.2% of the traces. This behavior is shown in Figure 3. For Permit Log, the model provided a process model for 40.62% of the traces. This behavior is shown in Figure 3.

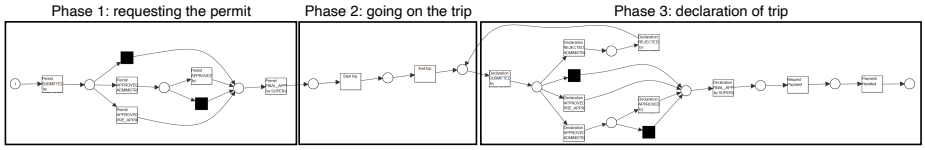

Figure 2: This behavior cluster, visualized as a Petri Net, explains 53.2% of the behavior seen in the International Declaration event log.

Phase 1 starts with submitting the permit. After the permit is submitted, it is either pre-approved or approved by the administration. If pre-approved, the supervisor can give final approval directly. Otherwise, it has to be approved by the supervisor first. In phase 2 the traveler goes on the trip. The start and the end of the trip are pre-recorded on the permit document. After the trip, phase 3 starts. Here, the traveler submits their travel declaration. This declaration is then rejected, pre-approved, or approved by the administration. If rejected, the traveler can submit his declaration again. If pre-approved, the supervisor can give his final approval directly. If approved, the declaration can be approved by the supervisor before getting final approval, but can also get final approval by the supervisor directly.

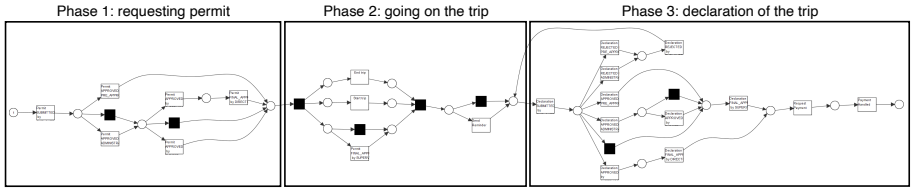

Figure 3: This behavior cluster, visualized as a Petri Net, shows 40.62% of the behavior seen in the `Permit Log` event log.

Similar to Figure 2, this process has three phases with similar activities. The difference in this process model is that because more behavior is similar to other traces, more behavior is shown in the process model. The only additional activities are those that reject either the permit request or the declaration. This is because this cluster contains more behavior of this sort. The "happy flow" (the process steps with the least amount of resistance) is this present and remains the same.

4 Business questions analysis

Now that the event logs are chosen and initial insight of the attributes and control flow are gained, The questions asked by the TU/e can be answered. Note that these questions are answered based on unfiltered data and that outliers have been accounted for by providing the mean and median thereby understanding the skewness of the data. These questions are all answered based on the cases in `International Declaration` and `Permit Log`. To answer the questions, both event logs have been imported in the process mining tool Disco and in R.

1. **What is the throughput of a travel declaration from submission (or closing) to paying?**

   The global statistics feature in Disco provided the mean and median throughput time. The mean throughput time (the time it takes a single case to be completed from start to finish) of a travel declaration from submission to paying is 10.1 days, while the median is 14.2 days.

2. **Is there a difference in throughput between national and international trips?**

   The global statistics feature in Disco provided the mean and median throughput time. The mean throughput in `International Declaration` is 10.1 days, with a median of 14.2 days. Compared to `Domestic Declaration` with a mean throughput of 11.5 days and a median of 7.3 days. This means that the throughput in `International Declaration` is, on average, 40.5% longer.

3. **Are there differences between clusters of declarations, for example between cost centers/departments/projects, etc.?**

   Filtering `OrganizationalEntity` in Disco on frequency, the five departments that declare most often are identified. These five departments account for 73.51% of the total recorded cases. By selecting each department individually and looking at the global statistics, the mean and median throughput is identified, see Table 1.

Department	**Mean (in days)**	**Median (in days)**	**Relative frequency (in %)**
65458	81.0	62.6	21.60
65455	81.4	62.2	15.69
65456	75.7	68.2	14.93
65454	77.1	18.5	13.17
65459	73.2	65.3	8.12

Table 1: Mean and median throughput time of declarations for the top five departments

4. **What is the throughput in each of the process steps, i.e. the submission, judgement by various responsible roles, and payment?**
   To get the throughput time of each process step, irrelevant activities (i.e. activities that do not belong to that process step) were filtered in Disco. In this way, each perspective is individually analyzed. The global statistics feature provided the following mean and median throughput, see Table 2:

Process step	Mean	Median
Submission	59.1 hours	0 seconds (most traces have no duration for `concept:name`
Judgement	66 hours	5.5 days
Payment	3.2 days	3.5 days

Table 2: Mean and median throughput time of the process steps

5. **Where are the bottlenecks in the process of a travel declaration?**

   Using the replay function in Disco, the bottlenecks can be visually identified. With the function, the biggest bottle neck seems to be getting initial approval from the administration, final approval from the supervisor, and handling the payment. In Figure 4, the replay function is visualized. Here we see the travel declarations of `International Declaration`. The green and red balls on the arcs represent a single case. The bigger the ball, the more cases are waiting for that event to complete.

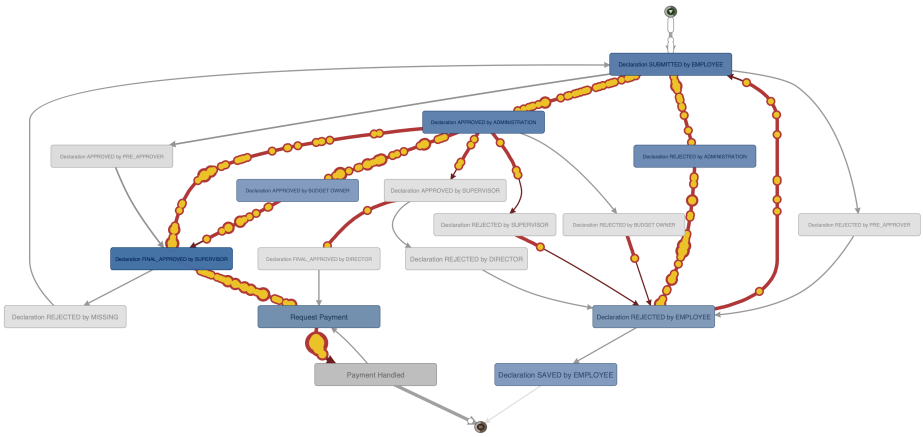

Figure 4: Disco's replay function of `International Declaration` travel declarations

6. **Where are the bottlenecks in the process of a travel permit (note that there can be multiple requests for payment and declarations per permit)?**

   The same approach was used for this question as for question 5. Replaying `Permit Log` showed that the biggest bottleneck is getting the permit approved by the budget owner and getting final approval by the supervisor, see Figure 5.

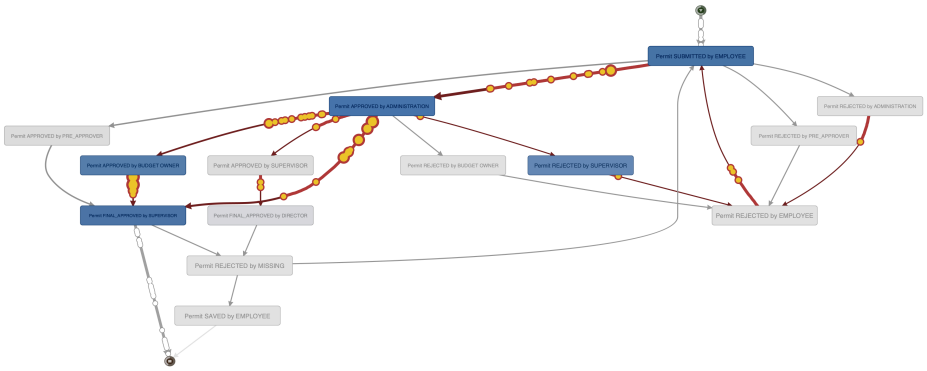

Figure 5: Disco's replay function of `Permit Log` permit requests

7. **How many travel declarations get rejected in the various processing steps and how many are never approved?**

   First, filtering in Disco was applied to only showcases that contain a "reject" event at any point in time. The global statistics feature showed that 1,576 cases remained. To know how many cases were never approved, all events leading to payment were filtered out. This showed that in total, 171 cases were never approved. The frequency per processing step is shown in Table 3. Here, absolute frequency shows how many times the event occurred and case frequency shows for how many different cases the event occurred.

Rejected by	**Absolute frequency**	**Case frequency**
Employee	1.780	1.483
Administration	1.549	1.287
Supervisor	126	122
Missing	103	98
Pre-approver	84	82
Budget owner	40	40
Director	4	4

Table 3: Absolute and case frequency of event that reject a travel declaration.

8. **How many travel declarations are booked on projects?**

   By filtering all cases that did not have a project number in Disco, 4116 out of 6752 cases remain. This means that 2636 cases are not booked on a project.

9. **How many corrections have been made for declarations?**

   Corrections are identified as cases that are at one point rejected but in the had a payment handled. This means that after the rejection they have corrected their submission. Out of the 1.576 cases that are rejected at one point in time, 1.312 are eventually accepted and paid.

10. **Are there any double payments?**

    This question required a different approach. `International Declaration` is imported in R and filtered on the payment handled event. This left an event log that only contained the payment handled event. Next, the declaration number was checked for duplicates. With this approach, no duplicate payments were identified. The event log was also analyzed in Disco by checking whether any cases had two payments. No cases showed any absolute or case frequency of the event higher than 1, meaning there was no double payment. Moreover, there are cases where payment handled is followed by request payment. However, these requests are never honored. Thus, no double payments are found in `international declaration`. This does not exclude projects or travel declarations that received a double payment by declaring costs twice. However, the approach that was used excluded a payment being made twice within a single declaration. Searching for duplicate amounts of the handled payments showed 25 identical amounts. These might be double payments, but as the attribute information of all trace is different no conclusion can be drawn based on this information.

11. **Are there declarations that were not preceded properly by an approved travel permit? Or are there even declarations for which no permit exists?**

    In Disco, events were filtered based on what did or did not follow them. In `International Declaration`, 426 payments are handled without being followed by an approved permit. Fortunately, none of these payments had there permit rejected at any point in time, meaning that the payment was not wrongfully handled after a permit rejection. However, this does not mean there never was a permit. When importing the data into R and checking whether any attribute related to a permit was not available, no cases where found. Thus, no declaration was paid without a permit.

12. **How many travel declarations are first rejected because they are submitted more than 2 months after the end of a trip and are then re-submitted?**

    When filtering `International Declaration` on cases where a submission was followed by ending a trip and then re-submitted, only 1 case was identified. In fact, this re-submission is also rejected and eventually resubmitted

again. This could mean that the declaration was submitted more than 2 months after the end of a trip twice.

13. **How many travel declarations are not approved by budget holders in time (7 days) and are then automatically rerouted to supervisors?**

    First, Disco's filter is used to only show submissions that are followed by the approval of a supervisor. However, when looking at the performance perspective of Disco, there are no cases where a declaration submission is followed by the approval of a supervisor after more than 7 days.

5 Data preparation

Now that the data has been described and most business questions have been answered, new insight can be sought in the data. First, data is prepared for the analyses. Before applying the prediction techniques, a significant amount of data cleaning and transformation is performed. First, the XES files were imported in Python using the PM4Py package. With PM4Py's `xes_import_factory` function, the XES files were loaded into an `EventLog` object before being exported using the `csv_exporter` into a comma-separated files (CSV) so the data could be imported into R for further preparation. During this process, the event log content was not changed in any way except for a "case." identifier at the beginning of the naming scheme to indicate it is a case attribute. From here on out, the attributes described in the data description are referred to as variables, as they are not in CSV format.

5.1 Column identification

As the origin of the data and the naming scheme of the variables are unknown, apparent combinations of unique identifiers between the files have been investigated. In this comparison, the aim was to look for identifiers that would connect one file to the other, joining the data. This means that the data of a declaration is included in the `International Declarations` and links to a permit in the `Permit Log`. However, it is unclear on which of the identifiers (i.e. the variables) in the former can be linked to the identifiers in the latter. Therefore, individual pairs were joined, after which the data was explored to look for the most complete result by looking at whether the `case.RequestedAmount` and `case.TotalDeclared` variables were similar in both files. For both files, columns are dropped that could not be interpreted without business context. For both files, the following columns were kept:

```
International Declarations:
id, org.resource, concept.name, time.timestamp, org.role,
case.Permit.id, case.RequestedAmount, case.Permit.Tasknumber,
```

`case.Permit.BudgetNumber, case.Permit.ProjectNumber,`
`case.Permit.OrganizationalEntity,`
`case.Permit.RequestedBudget, case.Permit.ActivityNumber,`
`case.AdjustedAmount, case.DeclarationNumber`

`PermitLog:`
`case.TotalDeclared, case.RequestedAmount_0, case.Overspent,`
`case.travel.permit.number, case.DeclarationNumber_0,`
`case.OverspentAmount,case.id`

5.2 Cleaning and preprocessing

The columns were renamed such that they did not have their prefixes (`'travelPermitNumber'`, `'declarationNumber'`, `'permitId'`) so that the variable names are easier to work with. For both files, some columns included their column name in the values themselves, such that the values in the table were, in case of the permit number, `'permit number xxx'`. These columns were changed by removing the text from these columns with a regular expression leaving only the numeric data. This transformation was done for the following rows:

`International Declarations:`
`taskNumber, permitBudgetNumber, permitActivityNumber,`
`organizationalEntity, permitId, projectNumber, declarationNumber`

`PermitLog:`
`travelPermitNumber, declarationNumber, permitId`

Filtering: With the choice to join the two sets by the *International Declarations* and joining that with the *Permit Log*, all rows where the declaration number was `N/A` were removed, only keeping cases where the information about the declaration is known. This filter reduces the number of records to be analyzed in the *International Declarations* set from 72151 to 70055, removing 2096 records.

Calculations: For *International Declarations*, additional columns were added that calculate the `process_duration` and `trip_duration` per declaration case instance. `process_duration` describes the time difference between the first and the last activity for each case. `trip_duration` describes the time difference between the `'Start trip'` and `'End trip'` activity for each case.

Joining: Both files were joined, using `permitId` and `declarationNumber` as primary keys with a left outer join, such that the information from the *PermitLog* was added to the list of *International Declarations*. The merged table had 70055 rows, just like the *International Declarations* table.

Filtering the joined table: In the merged table, not all records contain information. The `'overspent'` column contained particularly many `N/A` values. Since this is an important column during the analysis, all columns where overspent was `N/A` were filtered. This step removed 19475 rows, leaving 50580 rows in the final table. As visible in the box plot (see Appendix A), the data consists of many outliers.

Outliers: Additionally, 3471 trips were completed instantly, with a duration of exactly 0 days. Therefore, trips with a duration longer than 50 days or shorter than 0 days were removed from the table. This removed 783 outliers with a trip longer than 50 days and 3471 trips with a length of 0, leaving the table with 46316 rows. When visualizing the distribution of the continuous variables (`overspentAmount`, `process_duration`, `adjustedAmount`, `totalDeclared`, `requestedAmount`, and `permit.requestedBudget`), it showed large outliers. To remove these outliers, the column `overspentAmount` is filtered between its second and fourth quantile. This removed 2002 cases but produced much better distributions. In Appendix B, the distributions of each continuous variable is visualized. In this figure, the grey area represents the expected position for a normal distribution. Thus, by viewing Appendix B, we can conclude that none of the continuous variables are normally distributed.

Dummy variables: A separate table was created where the joined table was grouped on the `declarationNumber` and `permitId`. Thereby, each row represents a case, with its most important information, such as the process duration, trip duration, and overspent amount. Additionally, one-hot-encoding was applied to add columns for every possible activity, where each case has a 1 or 0 depending on whether the activity is present during the case. This was placed in a separate table as it removes information such as the sequentially and timestamps for the events. However, using this table, predictions can be made based on features (columns) to a target variable (a different column).

6 Analysis

This chapter describes the performed analysis, starting with an analysis to which variables might be correlated with each other, which leads to different prediction methods, and concluding in a hybrid approach when the data is analyzed through ProM.

6.1 Correlation analysis

As a first step in the analysis, the correlation between the different variables was investigated, in order to see how predictive they might be of each other. From the correlation matrix in Figure 6 several observations can be made:

Figure 6: Pearson correlation matrix between the variables

1. `Overspent amount`: The amount of money overspent seems to be very weakly correlated to the other variables requested amount, process duration, trip duration & total declared) except for the requested budget from the permit, where it shows a moderate negative correlation, indicating that if the requested budget from the permit increases, the average amount of overspending decreases.
2. `Requested amount`: The amount of money requested from the international travel declarations shows positive correlations to most other variables, but its correlation to overspent amount is, again, very weak at 0.05. Interestingly, it has an almost perfect correlation to total declared, indicating that these values presumably represent the same underlying data, but vary slightly from different randomization applied during the anonymization process of the data for the BPI Challenge.
3. `Process duration`: The average process duration has a very weak correlation to the other variables, again with the weakest correlation to the overspent amount. This would indicate the process duration to be a hard value to predict based on the other available data.
4. `Trip duration`: The trip duration has a moderate correlation with the other variables, and again its weakest correlation is with the overspent amount.
5. `Total declared`: The amount of money declared has quite a high correlation with the requested budget at 0.7, and again an almost perfect correlation with the requested amount, indicating varying randomization. Its correlation with `overspentAmount` is, again, very weak.
6. `Permit requested budget`: The requested budget from the permits as a medium to high correlation with most variables, and therefore could provide

some variance from the other variables. For both total declared and requested amount, the correlation is the same at 0.7, which is not surprising since these two variables have an almost perfect correlation between each other.

To further analyze the relationships between the different variables, a Principal Component Analysis (PCA) was performed through R's `prcomp` package, and visualized in a biplot from `ggbiplot`, seen in Figure 7. The plot on the left shows the PCA with all data points plotted, and the right plot shows the data points at 0.005 transparency so the directions are readable.

Figure 7: Biplot based on PCA

In PCA, variables are analyzed with regards to their contribution towards the variability in the data. This is done by combining variables in 'principal components', where each principal component represents some percentage of explained variability in the data. In a biplot, PCA is visualized with arrows indicating the direction each variable contributes towards this variability within the principal components. In the X and Y axes, the percentage of explained variance is shown. The arrows represent the variable vectors within the space, where the length of the arrow shows the weight within the component. Additionally, if arrows are at a right 90° angle from each other, a change in one variable does not influence the values in the other variable.

As leads from the PCA, overspent amount is at an almost right 90° angle from requested amount and total declared. This insignificant relation can also be seen in the correlation matrix: their correlations are only 0.05 and 0.06 respectively. Trip duration and process duration are very close to the origin, indicating that

their weight within the component is quite low. Permit requested budget has the most significant weight, especially within the first principal component. With a correlation to overspent amount of -0.41, this should be an interesting variable for further prediction problems.

6.2 Feature selection

With insight into the relations between the variables, the next step is to utilize feature selection methods to see which variables have predictive power. For this, R package `glmnet` is utilized to calculate the lasso feature regularization. First, through `cv.glmnet` with 100 folds, the optimal lambda is calculated, in this case, the lambda is set to 1.125, as shown in Figure 8. From the subsequent `glmnet` plot, Figure 9, the coefficients are plotted against their respective L1 norms. As leads from plugging the minimal lambda value to `glmnet`, there are four variables with significant predictive power with respect to overspent amount: `process duration`, `trip duration`, `total declared`, and `permit requested budget`. In addition to statistical predictive power, common sense needs to be taken into account: in order for analyses to be useful, only values that are known should be considered. Therefore, the process duration, which is only known after process completion, was excluded for further analysis. This leaves `trip duration`, `total declared`, and `permit requested budget` as the selected features for further prediction.

6.3 Model evaluation

In the next two sections, prediction techniques are described to analyze influences on overspent amount. These techniques are evaluated in the same way. First, a train and a test set are created with a 9-to-1 distribution: 3595 rows in the train set, and 400 rows in the test set, randomly sampled. After the models are trained on the train data set, the values for the test set are predicted. Since overspent amount is a continuous variable, the results are not trivial to evaluate: the predictions are very rarely 100% accurate because of the noise in the data set. Therefore, two metrics are utilized to evaluate the predictions:

- The difference between the prediction and the true value, which can be represented by the mean and standard deviation of this column. This gives the mean error of the prediction based on the test set.
- The percentage of predictions that are accurate to €100, which gives an idea of the reliability of the predictions and the usability in practice.

6.4 Linear prediction

To benchmark further predictions, linear regression is applied as a first step. Linear regression is a transparent, easy to understand and fast prediction algorithm, and is commonly used as a benchmark to compare more complex algorithms. Using `lm` from `Base R`, the selected features are used to predict overspent amount, see Table 4.

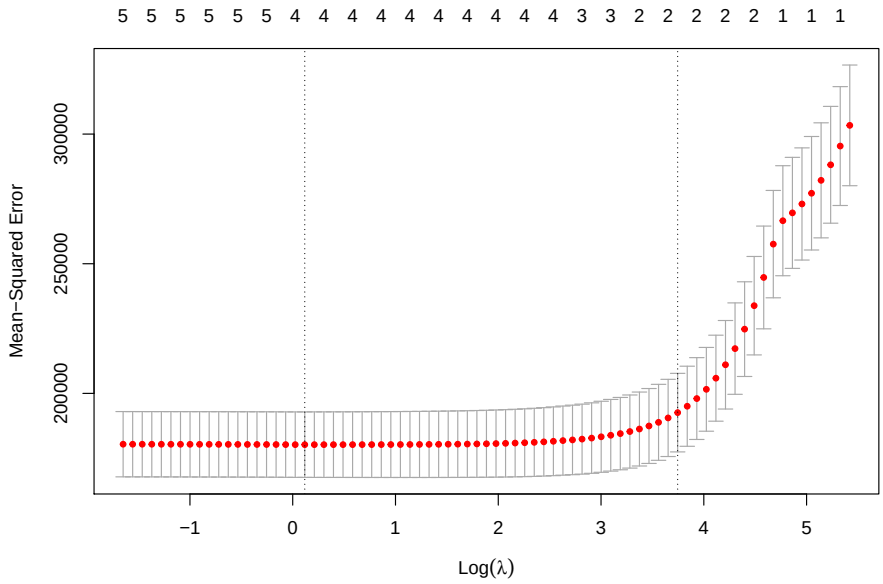

Figure 8: Lasso feature regularization

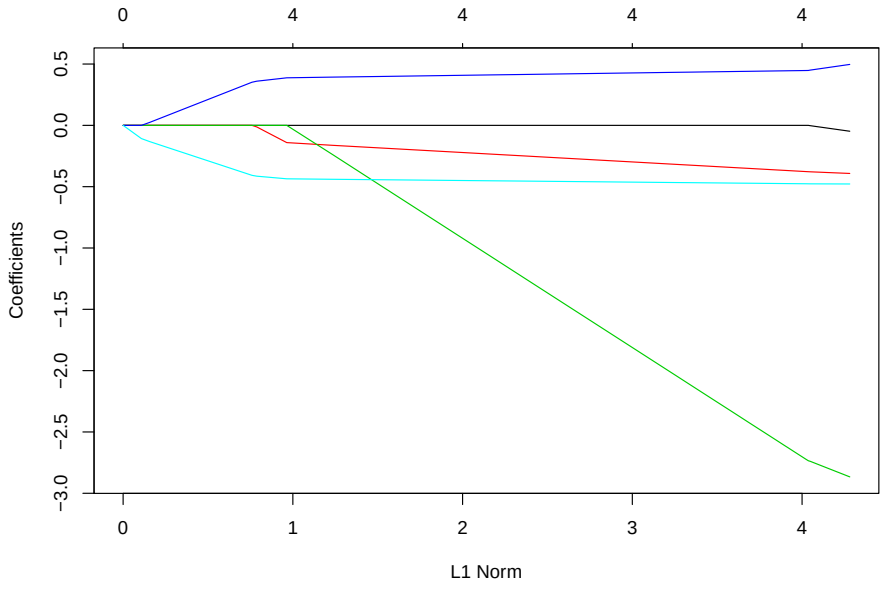

Figure 9: GLMNET coefficients

Features	Within €100	Mean difference	SD difference
Trip duration, total declared, permit requested budget	73%	26.85	418
Trip duration, total declared	75.5%	25.25	543.5
Trip duration, permit requested budget	69%	37.7	496
Total declared, permit requested budget	72.8%	26.47	419

Table 4: Linear regression feature comparison

6.5 AutoML

With the features selected and knowing how to benchmark the results, more advanced techniques can be applied to see if they can improve upon these results. For this, machine learning is utilized. Within machine learning, various different algorithms can be applied for prediction purposes. Some of these are already established algorithms, such as decision trees, Naïve Bayes, or Support Vector Machines. However, comparing algorithms is not only time-intensive, but different algorithms also require tuning of different hyperparameters, and manually programming these algorithms expertise and time. This makes advanced machine learning techniques hard to access for non-experts. To combat this steep learning curve, researchers and industry have come up with many different approaches. One such approach is AutoML, or Automated Machine Learning.

First hinted at by [6], by automatically comparing algorithms and then automatically generating hyperparameters for promising algorithms, the entry barrier for machine learning is lowered significantly. In this way, only an X (distribution) and Y (target variable) need to be provided, from which an automated machine learning approach can automatically select and optimize the prediction problem. For this project, `H2O` was chosen as platform because of its accessibility and compatibility with the analysis within `R` [5]. H2O automates both the comparison of the different algorithms as well as the tuning of hyperparameters up until the stopping points. Finally, the leading algorithms from the training phased are combined into a *stacked ensemble*, which typically performs better than the other algorithms because it is optimized for the training set [5].

In `H2O`, only 4 parameters need to be provided: the X and Y, the training frame from `H2O`, and a stopping condition, i.e. a number of models or a maximum run-time per model. For the purposes of comparison, these models do not need to be perfectly tuned, and therefore 100 models were trained, with a maximum run-time per model of 240 seconds. `H2O` was used in two separate instances with different features. In total, two instances were run with the best features of the benchmark linear prediction model: `Trip duration`, `Total declared`, and `Permit requested budget`. Instance 1 used all three selected features while instance 2 used only `Trip duration` and `Total declared`. Both instances had a budget of 4 hours (240 seconds per model for a total of 100 models) to give

the algorithms enough time to tune. After both instances depleted the budget the following leaderboards were retrieved, see Table 5 and Table 6:

Model ID	rmse	mse	mae
StackedEnsemble_AllModels_AutoML_20200618_225328	411	168949	248
StackedEnsemble_BestOfFamily_AutoML_20200618_225328	411	169001	248
DeepLearning_grid__1_AutoML_20200618_225328_model_1	413	170711	251
DeepLearning_grid__1_AutoML_20200618_225328_model_7	413	170999	255
DeepLearning_grid__1_AutoML_20200618_225328_model_4	414	171471	249
DeepLearning_grid__1_AutoML_20200618_225328_model_17	414	172172	251

Table 5: Leaderboard for instance 1

Model ID	rmse	mse	mae
StackedEnsemble_AllModels_AutoML_20200619_034754	503	253618	295
StackedEnsemble_BestOfFamily_AutoML_20200619_034754	502	252375	296
DeepLearning_grid__1_AutoML_20200619_034754_model_4	503	253064	293
DeepLearning_grid__1 AutoML_20200619_034754_model_6	503	253618	295
DeepLearning_grid__1_AutoML_20200619_034754_model_5	203	253699	295
DeepLearning_grid__1_AutoML_20200619_034754_model_3	503	253784	294

Table 6: Leaderboard for instance 2

The leaderboards show that, for both instances, ensembling all models created by the instance yielded the best results. When the instance creates an ensemble of all models, it uses all algorithms to produce a single predictor. Hence, the ensembled model is used to predict on a test set, see Table 7.

Instance	Features	Within €100	Mean difference	SD difference
H2O 1	Trip duration, total declared, permit requested budget	76%	-24.78	407
H2O 2	Trip duration, permit requested budget	74%	9.83	491

Table 7: AutoML result comparison

6.6 Result comparison

The results show the AutoML offers a slight improvement over the best benchmark baseline set by linear regression, see Table 8. Instance 1 shows a .5% improvement for predicting within €100 of the true value, an absolute mean difference of 0.47, and a standard deviation difference of 136.5. These results show

that H2O matches the results of the benchmark linear regression, but improves upon these results through a lowered amount of variance.

Algorithm	Features	Within €100	Mean difference	SD difference
Benchmark	Trip duration, total declared	75.5%	25.25	543.5
H2O	Trip duration, total declared, permit requested budget	76%	-24.78	407

Table 8: Automated machine learning feature comparison

6.7 Process behavior comparison

Besides predicting, another indicator of when a case might overspend could be its behavior. Hence, in this last analysis, the combined event log is separated based on whether on `overspent` is true or false. This produces two event logs: `overspent_events` and `non_overspent_events`. The two events logs have 3995 cases. After separating, `overspent_events` had 1295 cases and `non_overspent_events` 2700.

The behavior is compared in two steps. First, the frequency of specific events is compared between the event logs. This could show differences in how the processes are executed. Differences that are spotted could indicate that the behavior more frequently leads to overspending. For the process comparison, the `Process Comparator` plug-in in ProM 6.9 is used. This produced Figure 10. The plug-in takes two inputs: group A and B. In Figure 10, group A is represented by `non_overspent_events` and group B by `overspent_events`. The output of the plug-in is a directly-follows-graph where the events are colored based on frequency discrepancies and where arrows and borders represent the combined frequency of that event of arc. The shade of the color represents the weighed frequency discrepancies when the logs are compared (i.e. if the event if dark blue is occurs significantly more frequent in `non_overspent_events` and if the event is dark red if occurs significantly more frequent in `overspent_events`). Moreover, all elements below a frequency threshold of 5% were removed to enhance readability.

For this comparison, the border and arc frequency is ignored, as it does not show a difference between the logs. This leaves the colored events. In Figure 10 we see that in `non_overspent_events`, 10.45% of the cases start with the events `'Permit APPROVED by SUPERVISOR'` and `'Permit FINAL_APPROVED by DIRECTOR'`, while in `overspent_events`, only 3.94% of the cases start with these events. The figure also shows that the permits of cases in `overspent_events` have to be approved by the budget owner and supervisor 6.51% more often. This means that the permits of cases that did not overspend were about 2.6 times more likely to receive final approval of a director. Besides this, the figure also shows that the declarations of cases in `overspent_events` were more likely to be approved by multiple resources and

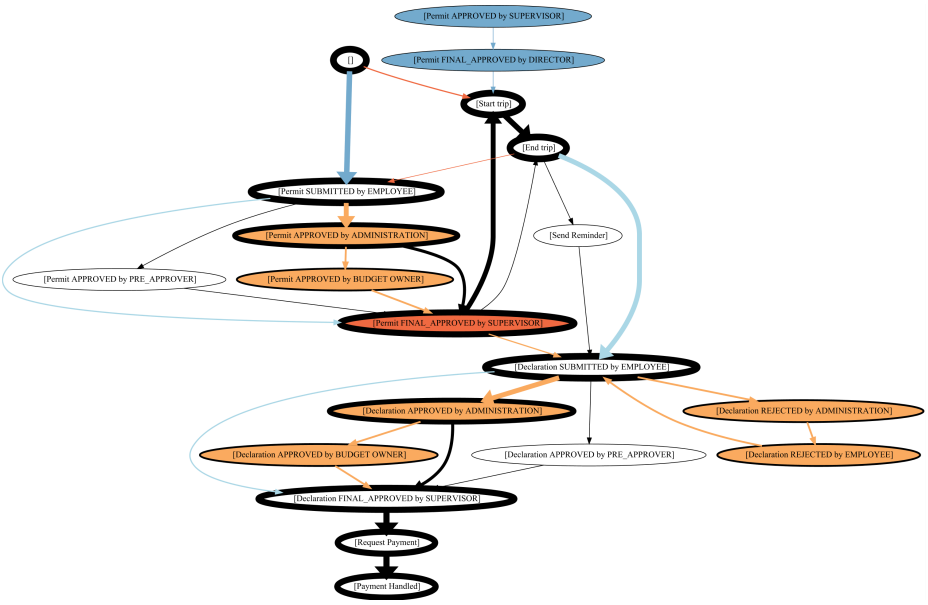

Figure 10: Comparison of `non_overspent_events` and `overspent_events`

more likely to be rejected before being paid. Moreover, even though cases that received final approval from directors are less likely to overspend, the process duration is also longer. Traces that contain the event `'Permit FINAL_APPROVED by DIRECTOR'` have a mean process duration of 73.6 days, while cases without have a mean process duration of 54.3 days, an increase of 35.55% (cases for which the process duration was more than three months have been removed to get a better comparison). Besides this, no further difference in behavior is identified based on the frequency of the events in both event logs.

this section, possible limitations and impact of this work is discussed. The limitations will focus mostly on the technical aspects of this report, while the impact of this work will mainly consider the ramifications of using an AutoML approach for hyperparameter optimization in process mining.

6.8 Limitations

In this report, business questions are answered and a novel AutoML approach for hyperparameter optimization in process mining is introduced. Answering the business questions did not lead to many limitations besides the lack of the inherent business context due to the unavailability of information. However, the AutoML approach did lead to several limitations. The AutoML approach shows that within process prediction, algorithm and hyperparameter optimization is a useful tool to enhance the performance of the prediction. However, it also

showed that it is difficult to intertwine traditional data mining techniques with process mining techniques. The difficulty was particularly apparent when trying to retain as much information about the original event log as possible while also fitting the data to the need of the AutoML approach. Moreover, the results of AutoML approach could potentially be improved by additional hyperparameter tuning and providing the model training phase with longer run time.

There are also several limitations with regard to the comparison of the behavior. Because only one technique is utilized to compare the behavior between the event logs, the analysis is not exhaustive. Additionally, only the frequency of the events is considered in the analysis, ignoring additional information about the process.

Lastly, because the provided event logs were randomized in an attempt to guarantee the anonymity of TU/e employees, analyses were hard to definitively validate. This could mean that the predictions are optimized based on generated data, and the model will most likely have varying results in practice.

6.9 Impact

This report could have an impact on both the TU/e and the broader scientific community. For the TU/e, the answers to the questions posed by the TU/e team in the BPI challenge could provide useful input for optimizing and improving the declaration processes. Additionally, the possibility of prediction the overspent amount could lead to a more targeted form of auditing and budgeting, where cases with a high risk of overspending could be analyzed further or preventive measures could be put in place. Lastly, the comparison between the overspent and non-overspent process variants shows that certain behavior is more often associated with overspending cases. This behavior could be further analyzed to uncover the root cause of this behavior.

For the scientific community, this report showed a, as far as the authors are aware, novel approach to utilize AutoML within the context of process mining. Beyond serving as a proof of concept, the implementation of AutoML was able to improve the results of the benchmark linear regression, while being simple to implement in a process mining problem setting. Moreover, the integration of this technique could further enhance the maturity of contemporary process mining solutions.

7 Conclusion and future work

This project proposes a novel approach to automated prediction optimization with an AutoML approach. Using the `H2O` platform, a model was produced that improved upon the results of a benchmark linear regression model. In this way, the report goal was achieved, and the degree to which a declaration is likely to be overspent can be estimated to within €100 in 76% of cases. Using an AutoML approach further reduced the variance in the prediction error by ±25%. Additionally, two process variants were compared to look for differences between

process behavior when a case is overspent. This comparison showed that if a case is approved by the director it is 2.6 times more likely not to overspend. Moreover, many of the business questions asked by the TU/e have been answered, providing business value for the TU/e to improve their processes.

This project provides several directions for future work. For one, the utilization of AutoML within process mining has not yet been demonstrated in a real-world case, and its usability within such a case is unknown. Additionally, future work could focus on applying AutoML to sequential data, instead of the case-based attributes demonstrated in this work. This could provide better results while retaining more information about the original event log.

Acknowledgements We would like to thank Jens Gulden and Xixi Lu for their help throughout the project. Additionally, we would like to thank to BPI challenge 2020 and the TU/e for providing the data and hosting the competition.

References

1. Bozkaya, M., Gabriels, J., van der Werf, J.M.: Process diagnostics: a method based on process mining. In: 2009 International Conference on Information, Process, and Knowledge Management. pp. 22–27. IEEE (2009)
2. De Weerdt, J., Vanden Broucke, S., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. IEEE Transactions on Knowledge and Data Engineering **25**(12), 2708–2720 (2013)
3. van Eck, M.L., Lu, X., Leemans, S.J., van der Aalst, W.M.: Pm^2: A process mining project methodology. In: International conference on advanced information systems engineering. pp. 297–313. Springer (2015)
4. Günther, C.W., Verbeek, E.: Xes standard definition 2.0. Eindhoven University of Technology (2014)
5. H2O.ai: H2O AutoML (June 2017), http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html, h2O version 3.30.0.1
6. Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 847–855. KDD '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2487575.2487629, https://doi.org/10.1145/2487575.2487629
7. Verbeek, E.: Bpi challenge 2020 (Mar 2020), https://www.tf-pm.org/competitions-awards/bpi-challenge/2020

Appendix A Box plot of all continuous attributes

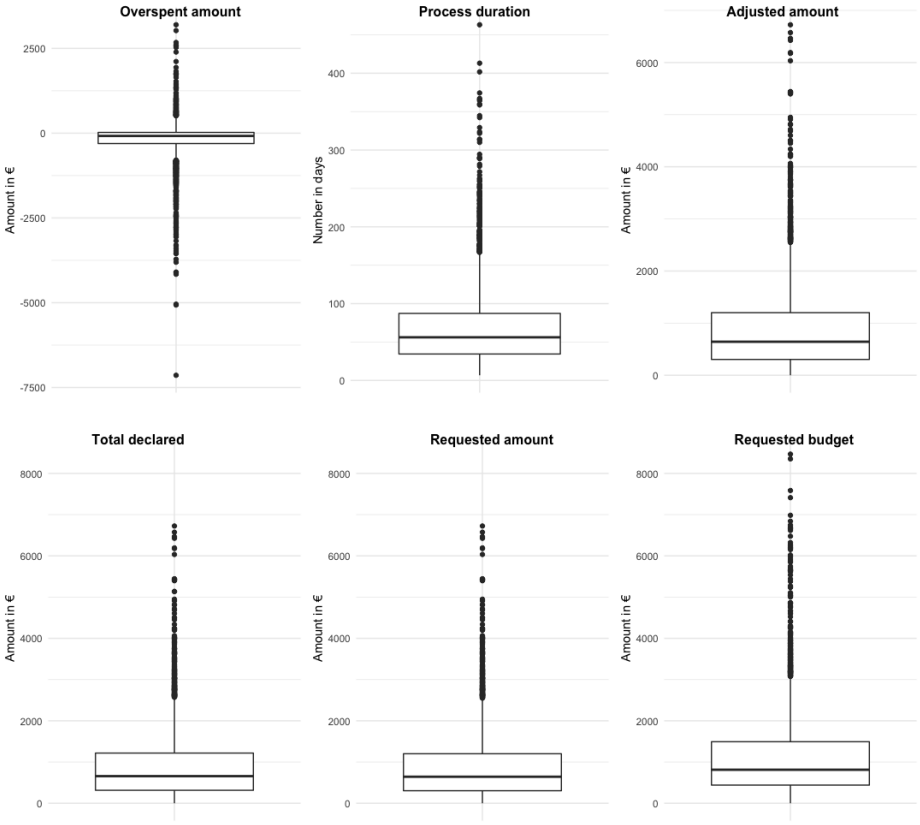

Figure 11: Box plot of all continuous attributes in the International Declarations and Permit Log event logs

Appendix B Distribution of continuous variables

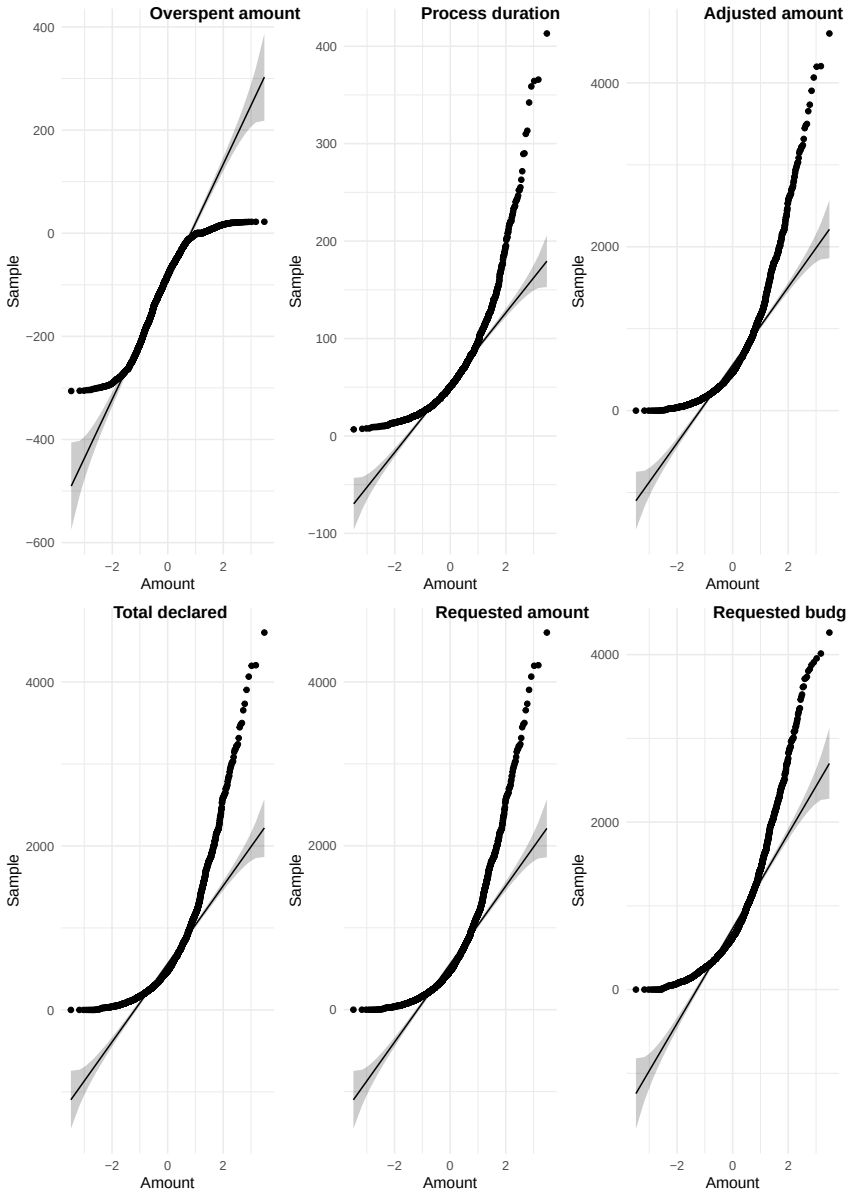

Figure 12: Distribution of continuous variables in the merged event log