

# Business Process Intelligence

## SS23 Assignment Part 2

Prof. Dr. Wil van der Aalst, Bianka Bakullari M.Sc., Harry Beyel M.Sc.,  
Benedikt Knopp M.Sc., Nina Graves M.Sc., Christopher Schwanen M.Sc.  
Chair of Process and Data Science  
RWTH Aachen University

June 1, 2023

---

## Introduction

In this assignment, you are provided with event data related to the customer support process of an online retailer called *BuyDirect*. BuyDirect is an e-commerce platform which offers a web service for online shopping. They sell a wide range of products across various categories, including electronics, fashion, home goods, beauty, sports and fitness, toys, and much more. BuyDirect places a strong emphasis on customer satisfaction and strives to provide a seamless shopping experience. Their goal is to offer secure payment options, a convenient and user-friendly interface, as well as fast responses to their customers' requests. The company has a dedicated customer support team whose goal is to provide assistance and solutions to any queries or concerns that arise throughout the shopping process.

The provided event data concerns events recorded in the context of customer support handled by the resources in the customer support team.

## The Data

Each case in the event data corresponds to a unique customer request handled by the customer support team. These requests can come from two types of customers: customers with *Free* subscription or customers with *Premium* subscription. In contrast to *Free* customers, customers with a *Premium* subscription pay a yearly fee which enables them early access whenever a new product line is launched. This information can be found in the case attribute *Subscription*. Upon arrival, each new request is registered. The resource who registers the request also determines the website component which may be responsible for causing the problem. This information can be found in the case attribute *Component* and it can have one of the following values:

- *Navigation*: This component encompasses all the elements and functionalities that allow users to explore and move through different sections and pages.
- *Authentication*: This component is responsible for ensuring safe customer login through credential validation while preventing unauthorized access.
- *Checkout*: This component covers the stage where the customers finalize their purchase and proceed to complete the transaction. It includes in particular the payment arrangement, shipping details and payment getaways.

- *Other*: Requests of this type concern problems that are not not clearly related to one of the other three components explained above.

Upon receiving a new customer request, it is promptly registered (activity: “Register”). If the issue at hand is evident and can be easily resolved, one of the team members addresses it immediately (activity: “Frontline Resolution”) without the need for further investigation. However, in most instances, a thorough investigation is required to comprehend the nature of the problem (activity: “Investigate”). Once the problem is identified and a suitable resolution method is determined, a comprehensive report is generated (activity: “Fill Report”), and a response is promptly sent to the customer (activity: “Send Answer”). When the customer acknowledges and accepts the response, the case is considered closed (activity: “Close”). Occasionally, customers may send additional emails or notifications to inquire about the current status of their existing request. Each of these subsequent interactions (activity: “Follow Up”) typically initiates a fresh investigation to ensure that the new inquiry is related to a previously acknowledged problem.

In addition to the event attributes *Case Id*, *Activity*, *Timestamp* and *Resource*, the event data also contains attributes *IT-Support* and *Estimate*. The *IT-Support* attribute refers to a member of the customer support team who is responsible for forwarding more complex tickets to another team responsible for the underlying component. The assigned IT support person remains the same throughout the same calendar week. On the other hand, the *Estimate* attribute represents an estimation of the hours required to complete the corresponding request. This estimate is determined each time a “Register” or “Investigate” activity occurs.

## Submission Details

- Total number of points obtainable: 100
- Group size: 2-3
- Input: this assignment PDF, data (*activity-table.csv*, *case-table.csv*, *request-handling.xes*, *request-handling-noise.xes*)
- Deliverables: one PDF report

## Tools

In this assignment we use *Celonis*, *Prom* and *RapidMiner*.

## Deliverables

Your submission must be in the form of a **single PDF report of at most 20 pages** containing your answers to the assignment questions. Please include relevant screenshots and figures to support your answers into the PDF file. The grading will be based solely on the written report. Make sure to include **the names of all group members on the title page**.

In the report, you should present your results and provide explanations when required. Doing so, **clearly indicate which answer belongs to which question**. Moreover, the report should be **self-contained** (i.e., it should not require reading material outside your submission). In addition, please mind the following criteria when preparing the report:

- Proper spelling, punctuation, readability
- Comprehensive structure

- Use of **adequate** visualizations for showing (aggregated) results or illustrating methods
- Figures have captions, axis have labels, diagrams have headers
- Figure quality (e.g., size, resolution and relevance)
- All figures, tables and similar are numbered and referred to in text

**Do not re-upload the event data we provided in your submission.** When answering the questions, document what you did and carefully describe and explain your results as required. Results from previous questions can (and should) be referred to, to improve your discussion and explanation. For questions 1, 5, 6 and 7, feel free to submit screenshots of Dashboards created in Celonis EMS (as shown in the instructions). Make sure that each component in the Dashboard has a name or a number and that one can easily link any snippet of PQL code to the corresponding component (if required).

## Question 1: Exploring the Event Data (16 points)

In this question, you want to explore the event data using various analysis components in *Celonis*, including the Process Explorer, Pie Charts, Column Charts and Histograms. Upload the data tables *activity-table.csv* and *case-table.csv* to *Celonis* and follow the Data Integration steps as explained in the lectures and instructions. More precisely, the *activity-table.csv* must be identified as the **Activity Table**, whereas the *case-table.csv* must be assigned to be its **Case Table**. Make sure you connect the two tables via the case identifier and load the data before you start the analysis.

- (a) (2 points) Use the Process Explorer component to create a Directly-Follows Graph (DFG) with activities as nodes. Move both sliders up (to 100%) so that the DFG shows all (seven) activities and arcs obtained from the data. Provide a screenshot of the DFG.
- (b) (12 points) For each of the following tasks, provide a screenshot of the corresponding component.
  - 1. (1 point) Using a Pie Chart component, visualize the distribution of the values for the case attribute *Subscription*.
  - 2. (1 point) Using a Pie Chart component, visualize the distribution of the values for the case attribute *Component*.
  - 3. (1 point) Using a Pie Chart component, visualize the distribution of the combined values for the case attributes *Subscription* and *Component* together.
  - 4. (1 point) Using a Column Chart component, show for each resource (x-axis), the total number of activities handled by that resource in the process (y-axis).
  - 5. (2 points) Using a Pie Chart component, visualize the distribution of the *IT-Support* responsibility over the resources throughout the different weeks. I.e., each pie portion shows the fraction of the weeks for which a particular resource takes over the *IT-Support* responsibility. **Provide the PQL code lines required for the dimension(s) and KPI(s) of the component.**
  - 6. (1 point) Using a Pie Chart component, visualize how the executions of activity “Register” are split among the corresponding resources.
  - 7. (1 point) Using a Pie Chart component, visualize how the executions of activity “Send Answer” are split among the corresponding resources.
  - 8. (1 point) Using a Pie Chart component, visualize how the executions of activity “Follow Up” are split among the corresponding resources.
  - 9. (1 point) Using a Pie Chart component, visualize how often the resource named *Jane* executes different activities.
  - 10. (1 point) Using a Pie Chart component, visualize how often the resource named *Liam* executes different activities.
  - 11. (1 point) Using a Histogram Chart component, visualize the number of occurrences for the throughput time in days. In the advanced options, select the Specific bucket count and set it to 10 (buckets). This will divide your values into 10 equal-width buckets.
- (c) (2 points) Describe and summarize the findings 200-300 words.

Total for Question 1: 16

## Question 2: Advanced Discovery (12 points)

In the following you would like to learn more on the different discovery techniques you learned in your BPI lecture. You find the event log *request-handling.xes* on the process of handling complaints and want to use it to foster a greater understanding for the different discovery techniques.

### State-based Regions

First you would like to experiment with state-based Regions in ProM. You decide to use the “Mine Transition System” plugin to create a transition system from the log which you use as an input for the plugin “Convert to Petri Net using Regions” subsequently. You assess the model quality using the “Multi-perspective Process Explorer” plugin.

- (a) (4 points) Apply State-based Regions Mining with different parameter settings until you obtain two different Petri nets, both of which satisfy the following two requirements: 1) each Petri net has a fitness value of at least 90 %, and 2) each activity in the event log appears as a transition label at least once. For each of the two models, provide a screenshot, their fitness (average fitness) and precision as well as a description on the way the models were mined and the parameters used.

### Inductive Mining

As you know from your lecture, the inductive miner identifies process behaviour by “cutting” the event log based on characteristic structures of the DFG. You can find an incomplete version of the process model for the provided event log in the figure below. You do not need to use ProM for the following questions but it is strongly recommended. Independent of your choice of tool, be sure to include a description of your procedure in your solution.

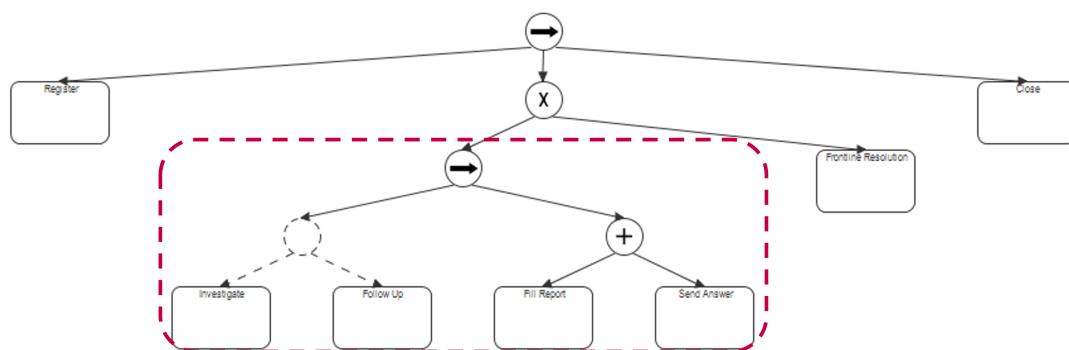


Figure 1: Process tree ( $M_1$ ) for the provided event log mined by the inductive miner.

- (b) (2 points) Use the provided event log to construct the event log the algorithm uses to detect the second sequence cut, so the event log including the marked sub-tree in Figure 1. Please provide the following information:
- Brief description of your procedure
  - Number of traces
  - Name of the included events
  - Number of variants

*Hint: To construct this log you have to perform two cutting operations on the log - you can see which ones in the process tree.*

- (c) (1 point) Now perform the sequence cut and create the resulting event log for the partition including the activities *Investigate* and *Follow up*. Please provide the number of traces and a screenshot of the all trace variants.
- (d) (1.5 points) Consider the trace variants and determine the missing cut-operator. Explain the reasoning behind your choice.

Consider the directly-follows graph in the Figure 2.

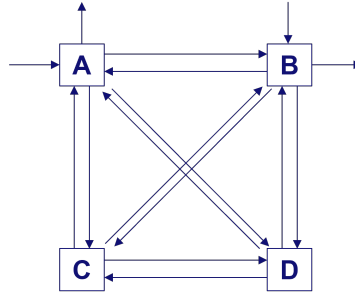


Figure 2: Directly-follows Graph

- (e) (1 point) What cut or cuts can be performed on the DFG in Figure 2?
- (f) (2.5 points) Provide the process tree the Inductive miner without log projection ( $IM_D$ ) would return for the DFG in Figure 2.

Total for Question 2: 12

### Question 3: Model Comparison (12 points)

So far you have learned a lot on how the different discovery algorithms work. Now you want to take a closer look at the quality of their results.

- (a) (6 points) Each of the four Petri nets below were created based on the result of a discovery algorithm on the event log you have been working with. One was mined with the Alpha miner, another with state-based regions. The remaining two are the Petri net representation of the result of the Inductive miner and a causal net mined using the Heuristic miner. Which of the Petri nets belongs to which discovery algorithm? Provide a reasoning for your choices based on the net's behaviour and/or its structure.
- (b) (3 points) Take a closer look at model  $M_4$ . Compute the percentage of traces from the event log which perfectly fit the model. Then, provide the footprint matrices of the event log and model  $M_4$  and based on them, compute the fitness.
- (c) (3 points) Can you modify  $M_4$  slightly in such a way that the model is able to replay more traces, but has a lower fitness based on the footprint matrix compared to  $M_4$ ? If so, provide the model, its footprint and both fitness values (replay and footprint); if not, explain why.

Total for Question 3: 12

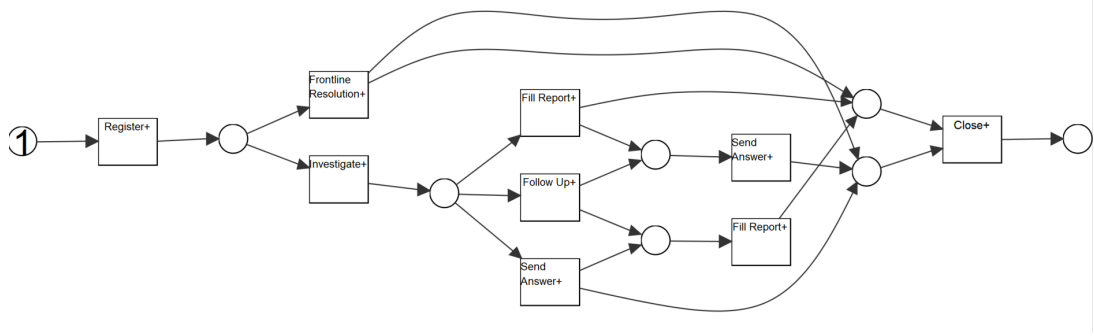


Figure 3: Petri net  $M_2$  determined using a discovery algorithm.

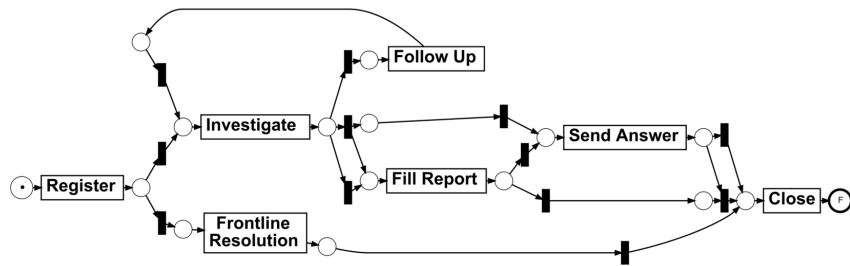


Figure 4: Petri net  $M_3$  determined using a discovery algorithm.

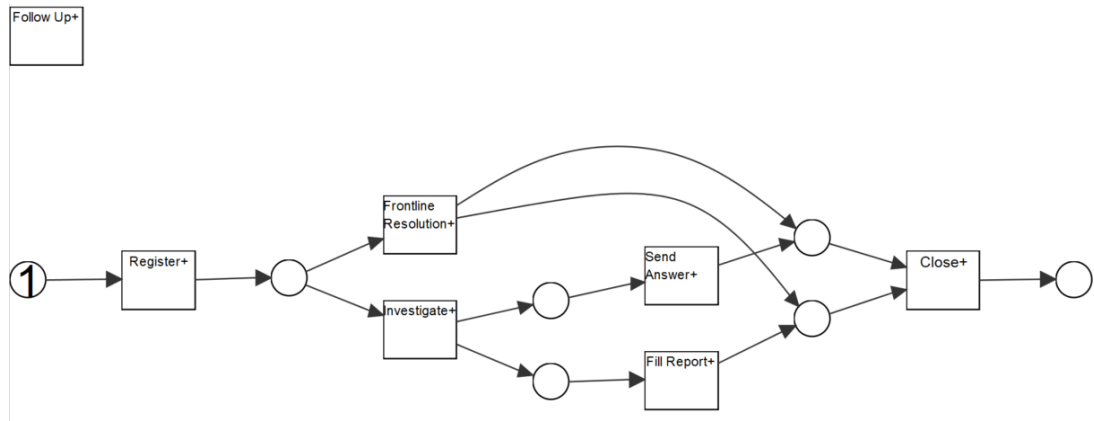


Figure 5: Petri net  $M_4$  determined using a discovery algorithm.

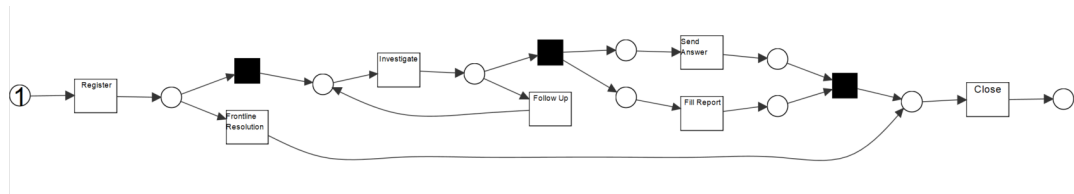


Figure 6: Petri net  $M_5$  determined using a discovery algorithm.



## Question 4: Conformance Checking (19 points)

The request handling system of *BuyDirect* had some issues lately and your manager asks you to run a conformance checking analysis on the recorded event log. Therefore, load the event log *request-handling-noise.xes* into *Celonis* and create an analysis workspace.

- (a) (1.5 points) Mine the reference model based on the most common trace variants that cover 50 % of cases. Add a screenshot of the model in your report. Also explain which of the process models  $M_2$ ,  $M_3$ ,  $M_4$ , and  $M_5$  from Question 3 appears most similar and why.

Please save the BPMN diagram on your computer for later use.

- (b) (8.5 points) Use the Conformance Checker in *Celonis* to write a report for your manager. How many traces, trace variants, activities, and events exist in the log? Describe the percentage of traces that perfectly fit the process model. Also contrast the full directly-follows graph of the perfectly fitting traces with that of the deviating traces.

In addition, inspect the reported violations. Which of these violations imply problems (just based on their message) that should not have occurred according to the process model? Inspect the cases belonging to these violations and suggest a better description of the deviation. Perform a token-based replay analysis for each violation's most common variant on the Petri net representation you have chosen in part (a) and use your results to elaborate on your suggestion.

*Hint: Considering three misleading violations here will be sufficient for your report.*

You want to be thorough with your report and try to incorporate further insights. Therefore, you import the event log into *ProM*. Save the process model you discovered and import it into *ProM* as well.

- (c) (5 points) First, investigate the event log with the visual Directly Follows Miner and set the path filter parameter to 0.7. In your report describe shortly each deviation you observe in the process model. Orientate your style on that of the violations in *Celonis* and refer to each arc indicating a deviation in a screenshot of the model.
- (d) (4 points) Replay the event log on the process model you mined before. Besides fitness, please also provide the values for precision and generalization. Use the plug-in "Convert BPMN diagram to Petri net (control-flow)" with the option "Translate for Conformance Checking" to obtain a Petri net from the BPMN diagram. Remove unnecessary silent transitions with the plug-in "Reduce Silent Transitions".

Finally, you get the information that the event log is affected by erroneous recording. In fact, not all events that actually happened are represented in the recorded data. Given this insight, how would you adapt the alignment computation? Describe your procedure and what has changed in the computed alignments. Provide the new values for fitness, precision and generalization in the adapted computation as well.

Total for Question 4: 19

## Question 5: Decision Mining (16 points)

In this task, the objective is to examine two specific decision points within the process and the contextual factors that may impact their respective outcomes. The first decision point involves determining whether a case is resolved upfront with the activity “Frontline Resolution”. The second decision point involves determining whether, following each investigation, a “Follow Up” request is made or if the case proceeds with activities “Fill Report” and/or “Send Answer.”

In the following, you must use *Celonis* to construct suitable situation tables which capture the contextual details together with the choices made at each decision point. The data and the data model you need for this task is the one you created in Question 1 (tables *activity-table.csv* and *case-table.csv*).

(a) (5 points) Create a *case-based situation table* containing the following columns:

1. Case identifier
2. Case subscription
3. Case component
4. Name of resource executing “Register”: This is the first activity for each trace.
5. First estimate category (*Simple*, *Normal*, or *Hard*): This value refers to the first value assigned to event attribute *Estimate*. The value must be *Simple* if the first estimate is lower than 165 (hours), *Hard* if the first estimate is higher than or equal to 675 (hours), and *Normal* otherwise.
6. Decision (*Resolution* or *Investigation*): This value should be set to *Resolution* if the case contains activity “Frontline Resolution”. Otherwise, the case runs through an “Investigate” activity and the decision value should be set to *Investigation*.

For columns 4, 5, and 6, provide the PQL code you use to compute them. Sort the rows in the table by the case identifier (in ascending order) and show the first 10-20 rows of the situation table (containing all columns 1-6).

(b) (2 points) Remove the case identifier column and download the case-based situation table you created in (a) containing columns 2-6. Create a decision tree in *RapidMiner* where you define column 6 (Decision) as the target variable, while columns 2-5 are the descriptor variables. Provide a screenshot of the decision tree produced using the Decision Tree operator with the following settings: set criterion to gain\_ratio, apply pruning, set confidence to 0.1, set minimal gain to 0.05, set minimal leaf size to 2, and maximal depth to 5.

(c) (1 point) Shortly describe two paths in the decision tree from the root to the leaves.

(d) (5 points) After each “Investigate” activity, the decision is made whether activity “Follow Up” or one of the activities “Send Answer” and/or “Fill Report” follows. Create an *event-based situation table* where each row corresponds to an “Investigate” activity. The table must contain the following columns:

1. Case identifier
2. Activity name: If you apply the correct filter to your table, this should always have the value “Investigate”.
3. The IT-Support resource
4. Resource executing the investigation
5. Number of currently running cases: This is the number of cases that have already started but not completed at the current timestamp.

6. Decision (*Follow* or *Solve*): This value should be set to *Follow* if the decision choice (upcoming activity) is “Follow Up”. Otherwise, the upcoming activity is “Send Answer” or “Fill Report” and the value must be set to *Solve*.

Provide the filter you apply to the table. For columns 4, 5, and 6, provide the PQL code you use to compute them. Sort the rows in the table by the case identifier (in ascending order) and show the first 10-20 rows of the table (containing all columns 1-6).

- (e) (2 points) Remove columns 1 and 2 and download the event-based situation table you created in (d) containing columns 3-6. Create a decision tree in *RapidMiner* where you define column 6 (Decision) as the target variable, while columns 3-5 are the descriptor variables. Provide a screenshot of the decision tree produced using the Decision Tree operator with the following settings: set criterion to *gain\_ratio*, apply pruning, set confidence to 0.1, set minimal gain to 0.05, set minimal leaf size to 2, and maximal depth to 5.
- (f) (1 point) Shortly describe two paths in the decision tree from the root to the leaves.

Total for Question 5: 16

## Question 6: Performance (17 points)

In this task, the objective is to examine process performance in relation to time. Specifically, you will conduct an analysis of the throughput time of cases, along with the waiting time between each investigation and a subsequent follow up request.

In the following, you must use *Celonis* to construct suitable situation tables which capture the contextual details surrounding the throughput and waiting times. The data and the data model you need for this task is the one you created in Question 1 (tables *activity-table.csv* and *case-table.csv*).

- (a) (1 point) Create two Single KPI components of type Number. One of them must compute the 0.3 Quantile value of the throughput times in **hours**. The other one must compute the 0.7 Quantile value of the throughput times in **hours**. Provide a screenshot of both components.
- (b) (4 points) Create a *case-based situation table* containing the following columns:
  1. Case identifier
  2. Case subscription
  3. Case component
  4. Decision (*Resolution* or *Investigation*): This is the same column as column 6 you computed in Task 5 (a). The value should be set to *Resolution* if the case contains activity “Frontline Resolution”. Otherwise, the case runs through an “Investigate” activity and the value should be set to *Investigation*.
  5. Solving order (*Report then Answer*, *Answer then Report*, or *Frontline case*): If the case executes activity “Frontline Resolution”, the value must be set to *Frontline case*. Otherwise, the case executes both activities “Send Answer” and “Fill Report” directly after each other in both possible orders. The value must be set to *Report then Answer* if “Fill Report” happens before “Send Answer” and it must be set to *Answer then Report* otherwise.
  6. Throughput time category (*slow*, *normal*, or *fast*): If the throughput time (in hours) is lower than the 0.3 quantile you computed in (a), the value must be set to *fast*. If the throughput time (in hours) is higher than or equal to the 0.7 quantile you computed in (a), the value must be set to *slow*. Otherwise, the value must be set to *normal*.

For columns 5 and 6, provide the PQL code you use to compute them. Sort the rows in the table by the case identifier (in ascending order) and show the first 10-20 rows of the situation table (containing all columns 1-6).

- (c) (2 points) Remove the case identifier column and download the case-based situation table you created in (b) containing columns 2-6. Create a decision tree in *RapidMiner* where you define column 6 (Throughput time category) as the the target variable, while columns 2-5 are the descriptor variables. Provide a screenshot of the decision tree produced using the Decision Tree operator with the following settings: set criterion to gain\_ratio, apply pruning, set confidence to 0.1, set minimal gain to 0.05, set minimal leaf size to 2, and maximal depth to 5.
- (d) (1 point) Shortly describe two paths in the decision tree from the root to the leaves.
- (e) (6 points) Create an *event-pair-based situation table* where each row corresponds to a pair of consecutive “Investigate” and “Follow Up” activities. The table must contain the following columns:
  1. Case identifier
  2. Source activity: This should always show the value “Investigate”.

3. Target activity: This should always show the value “Follow Up”.
4. Source resource: This should be the resource that executes activity “Investigate” for the current row.
5. Target resource: This should be the resource that executes activity “Follow Up” for the current row.
6. Follow ups until now: Activity “Follow Up” in the current row may have been executed multiple times. This value must be an integer indicating the current number of executions of “Follow Up”. E.g., it should be 3 if the current “Follow Up” is being executed for the third time within the corresponding case.
7. Waiting time category (*short wait* or *long wait*): If the time difference (in hours) between the source and the target event is lower than 15, the value must be set to *short wait*. Otherwise, the value must be set to *long wait*.

Provide the filter you apply to the table. For columns 4, 5, 6 and 7, provide the PQL code you use to compute them. Sort the rows in the table by the case identifier (in ascending order) and show the first 10-20 rows of the table (containing all columns 1-6).

*Hint: The filter should determine that each row in the table corresponds to a source activity “Investigate” and a target activity “Follow Up”. Moreover, you can look up the function `INDEX_ACTIVITY_TYPE` to compute column 6.*

- (f) (2 points) Remove columns 1, 2 and 3 and download the event-pair-based situation table you created in (e) containing columns 4-7. Create a decision tree in *Rapid-Miner* where you define column 7 (Waiting time category) as the target variable, while columns 4-6 are the descriptor variables. Provide a screenshot of the decision tree produced using the Decision Tree operator with the following settings: set criterion to *gain\_ratio*, apply pruning, set confidence to 0.1, set minimal gain to 0.05, set minimal leaf size to 2, and maximal depth to 5.
- (g) (1 point) Shortly describe two paths in the decision tree from the root to the leaves.

Total for Question 6: 17

## Question 7: Organizational Mining (8 points)

In this task, the objective is to analyze how resources work in this process. Specifically, you will determine clusters of resources that execute similar tasks, along with a social network showing how resources and resource groups hand over work to each other.

The data and the data model you need for this task is the one you created in Question 1 (tables *activity-table.csv* and *case-table.csv*).

- (a) (2 points) We want to analyze the resource-activity matrix to determine whether resources play different roles in the process. Create a *resource-based situation table* in *Celonis* using the Pivot Table component. Each row must correspond to a resource and each column must correspond to an activity. The table must contain all resources and activities. Each entry in the table corresponding to some resource  $r$  and activity  $a$  must show the average number of times per case that resource  $r$  executes  $a$  in the process.

Provide a screenshot of the Pivot Table. Moreover, for the Pivot Table, show the PQL commands you used for Dimensions and KPIs.

*Hint: Make sure that the formatting of the entry values in the Pivot Table is set to Decimal Number (#.##).*

- (b) (2 points) Export the resource-based situation table (the Pivot Table) you constructed in (a) and load it to *RapidMiner*. Run the k-means clustering algorithm using  $k = 3$ . What clusters of resources do you find? Describe which resources are clustered together and use the Centroid Table to describe their typical activities.
- (c) (2 points) Using the Process Explorer component, create a DFG where the nodes correspond to resources (instead of activities). You can achieve this by setting the Custom dimension of the component to the resource attribute of the activity table. In addition, change the node colors (Option Activity Colors) so that they reflect the clusters from (b) the resource belongs to. Note that there should be three distinct colors. Show a screenshot of the DFG.
- (d) (2 points) Using the Process Explorer component, create a DFG where the nodes correspond to the resource clusters. You can achieve this by putting a PQL formula in the Custom dimension which remaps resource values onto a string describing the cluster of the corresponding resource. Try to use meaningful cluster names (instead of e.g., cluster 0, 1, ...). Move both sliders to 100 %. Provide the PQL command used for the Custom dimension as well as a screenshot of the resulting DFG.

Total for Question 7: 8