Predictive Analytics

▼ Overview

Definition: Process mining prediction uses historical process data to forecast future events, durations, outcomes, or other aspects of business processes.

Purpose: The goal is to provide actionable insights for improving process efficiency, reducing risks, and optimizing resource allocation.

Key Components

- **Event Logs**: Historical data that captures the sequence of activities, timestamps, and other relevant attributes.
- **Predictive Models**: Algorithms and techniques (such as machine learning, statistical models, and heuristics) used to analyze event logs and make predictions.
- **Predictive Features**: Characteristics derived from event logs that serve as inputs to predictive models (e.g., event timestamps, activity types, resource usage).

Types of Predictions

1. **Outcome Prediction**: Forecasting the final outcome of a process instance (e.g., whether a loan application will be approved).
2. **Next Activity Prediction**: Predicting the next step or activity in a process instance.
3. **Remaining Time Prediction**: Estimating the time remaining for a process instance to complete.
4. **Resource Prediction**: Forecasting which resources will be needed at various stages of the process.

Techniques Used

- **Machine Learning**: Algorithms such as decision trees, random forests, neural networks, and support vector machines.
- **Time Series Analysis**: Techniques to analyze time-dependent data points and predict future values.
- **Statistical Models**: Regression analysis, probabilistic models, and other statistical techniques.

Process

1. **Data Collection**: Gathering event logs from information systems.
2. **Data Preprocessing**: Cleaning and transforming data for analysis.
3. **Feature Engineering**: Creating predictive features from raw data.
4. **Model Training**: Training predictive models using historical data.
5. **Model Validation**: Testing the accuracy and reliability of the models.
6. **Deployment**: Implementing predictive models in real-time process monitoring systems.

Challenges

- **Data Quality**: Ensuring completeness, accuracy, and consistency of event logs.
- **Complexity**: Handling complex and variable processes.
- **Interpretability**: Making predictive models understandable for business users.
- **Integration**: Combining predictive insights with existing business processes and systems.

Real-World Examples

- **Manufacturing**: Predicting machine failures to schedule timely maintenance.
- **Healthcare**: Forecasting patient outcomes to optimize treatment plans.
- **Finance**: Predicting loan defaults or fraudulent transactions.
- **Customer Service**: Estimating resolution times for customer support tickets.

Benefits

- **Proactive Decision Making**: Enables businesses to anticipate and mitigate issues before they arise.
- **Enhanced Efficiency**: Helps optimize resource allocation and process execution.
- **Risk Reduction**: Identifies potential risks and allows for preemptive action.

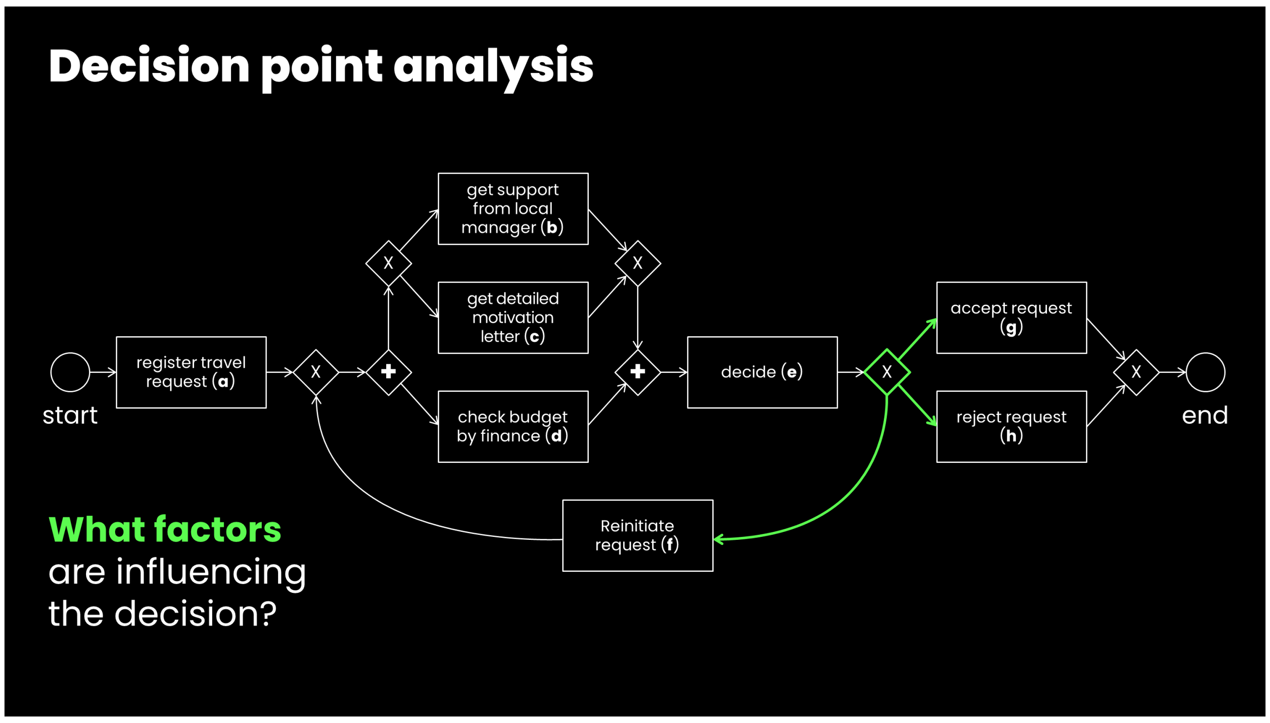
- **Customer Satisfaction**: Improves service delivery and customer experience by predicting and addressing potential delays or issues.

Process mining prediction bridges the gap between historical data analysis and future-oriented decision-making, providing organizations with a powerful tool to enhance their operational efficiency and strategic planning.

▼ Process Model Enhancement with Analysis

The focus thus far has been on *control-flow aspects*, e.g., discovering a DFG or BPMN model and checking compliance. However, by replaying event data on process models, it is also possible to *enrich* control-flow model using additional information hidden in event data. We will demonstrate this using a few examples.

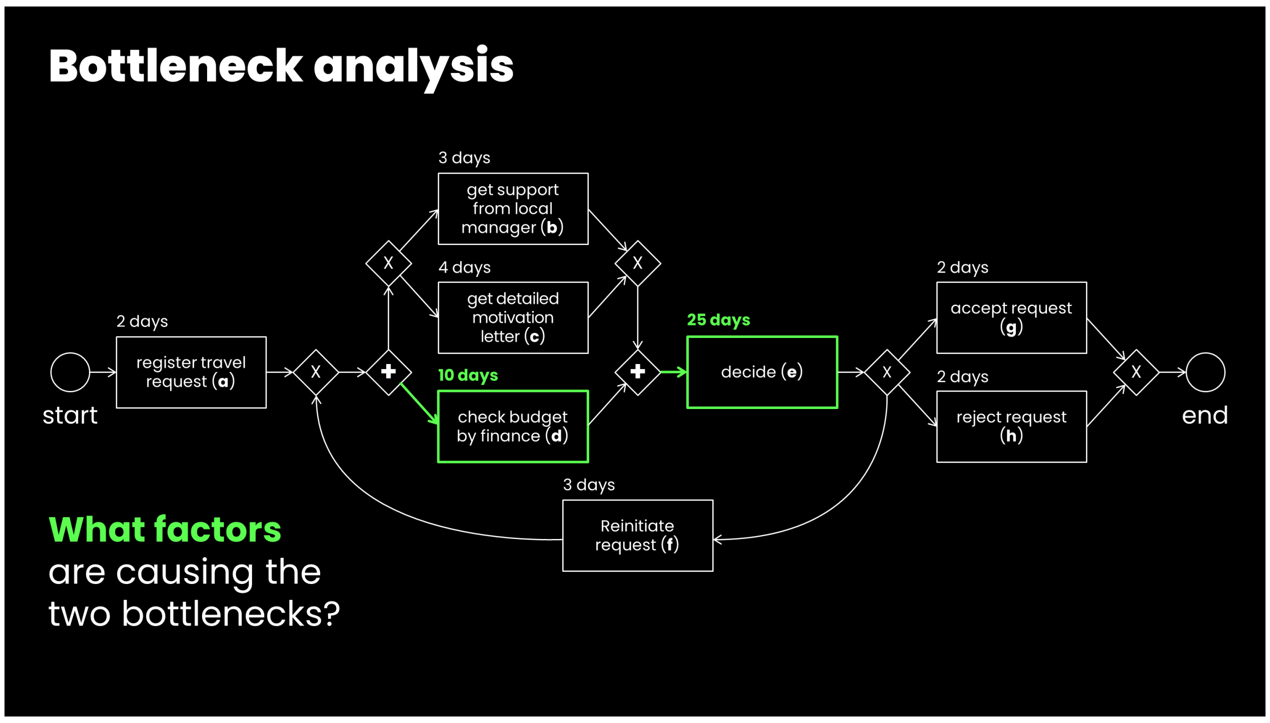
For every *choice* in the process model, it is possible to create a standard machine learning or data mining problem to learn what factors are influencing the choice. Each time a choice is made, the outcome is used as a target feature, and all other features are used as descriptive features, e.g., last activity, last resource working on the case, time of the week, workload, etc. Techniques ranging from decision-tree learning and logistic regression to neural networks and support vector machines can be used to explain the target feature in terms of the other features.


Decision point analysis

What factors are influencing the decision?

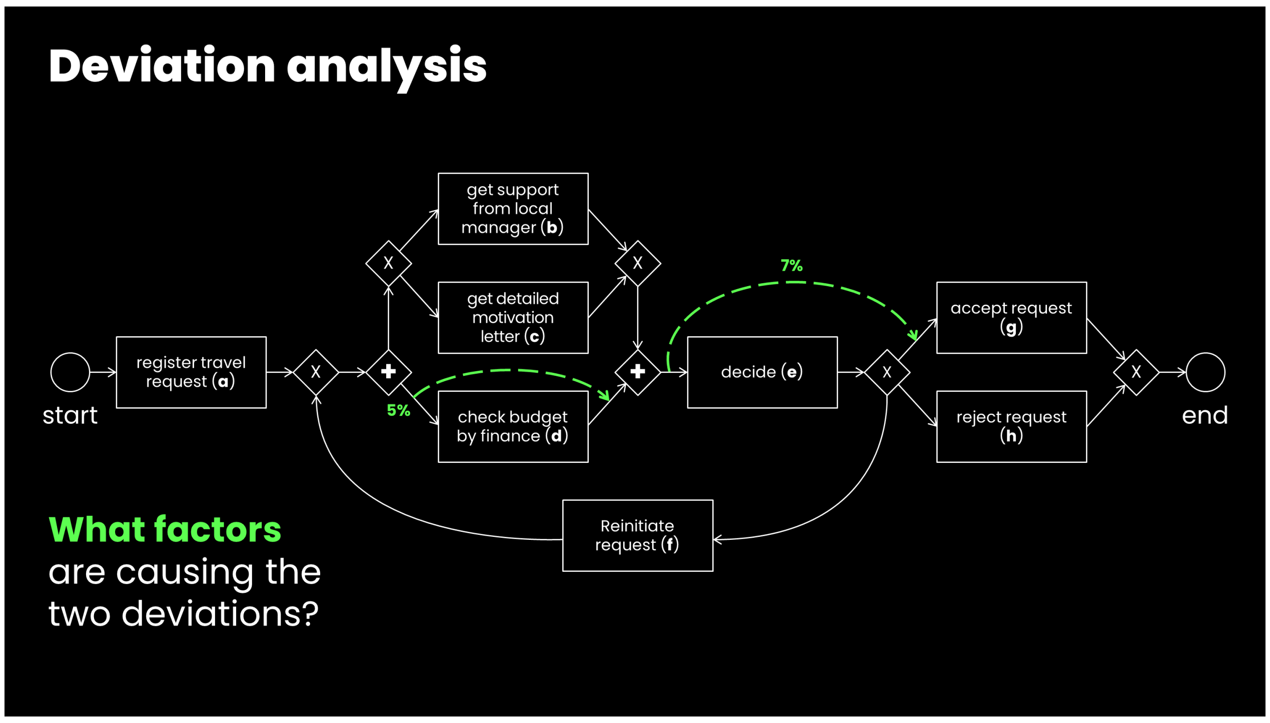
start
register travel request (a)
X
get support from local manager (b)
get detailed motivation letter (c)
check budget by finance (d)
decide (e)
accept request (g)
reject request (h)
Reinitiate request (f)
end

The same applies to delays in the process. Using the timestamps in event data, one can measure waiting times for activities, activity durations, and the time in-between selected activities. This way, we can annotate models with bottleneck information.


Bottleneck analysis

What factors are causing the two bottlenecks?

Using conformance checking, we can also annotate the process model with information about *deviations*. For example, it is possible to show how often mandatory activities are skipped.

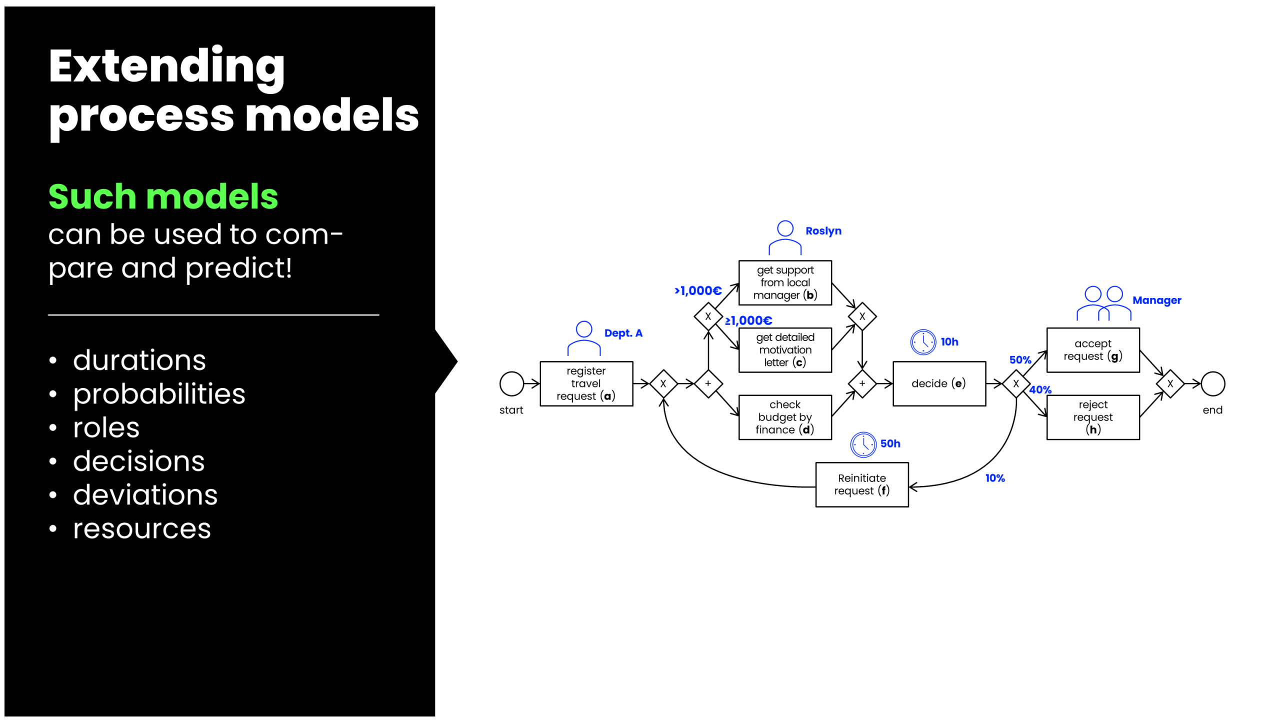


Deviation analysis

What factors are causing the two deviations?

start
register travel request (a)
X
+
X
get support from local manager (b)
get detailed motivation letter (c)
check budget by finance (d)
5%
X
+
7%
decide (e)
X
accept request (g)
reject request (h)
X
end
Reinitiate request (f)

In summary, process models can be extended with information about decisions, durations, probabilities, roles, resources, etc.



Extending process models

Such models can be used to compare and predict!

- durations
- probabilities
- roles
- decisions
- deviations
- resources

Next to the control flow perspective (i.e., the ordering of activities), process models may include other perspectives, including time, data, resources, costs, etc. For example, a choice may be based on the attributes of the case or preceding event, and we may attach resource allocation rules to activities (e.g., role information and authorizations). Timestamps and frequencies of activities can be used to identify bottlenecks. Process mining techniques may add such perspectives, but we typically try to get clarity on the control flow first. After adding the different perspectives, an *integrated* process model can be obtained.

Organizational mining focuses on the organizational perspective. This includes social network analysis to understand the role of resources (e.g., individuals, machines, or organizational units) in the process. Counting "handovers of work" is just one of many ways to construct a social network from an event log and a process model. The behavior of a resource can be characterized by a "profile", i.e., a vector indicating how frequent each activity has been executed by the resource. By using such profiles, various clustering techniques can be used to discover similar resources and thus identify roles. This information can be used to annotate process models with role information. Similarly, information on time, data, resources, costs, etc. can be added, resulting in a holistic view of the process. This provides new insights and may generate various ideas for process improvement. Moreover, the integrated model can be used as input for other tools and approaches.

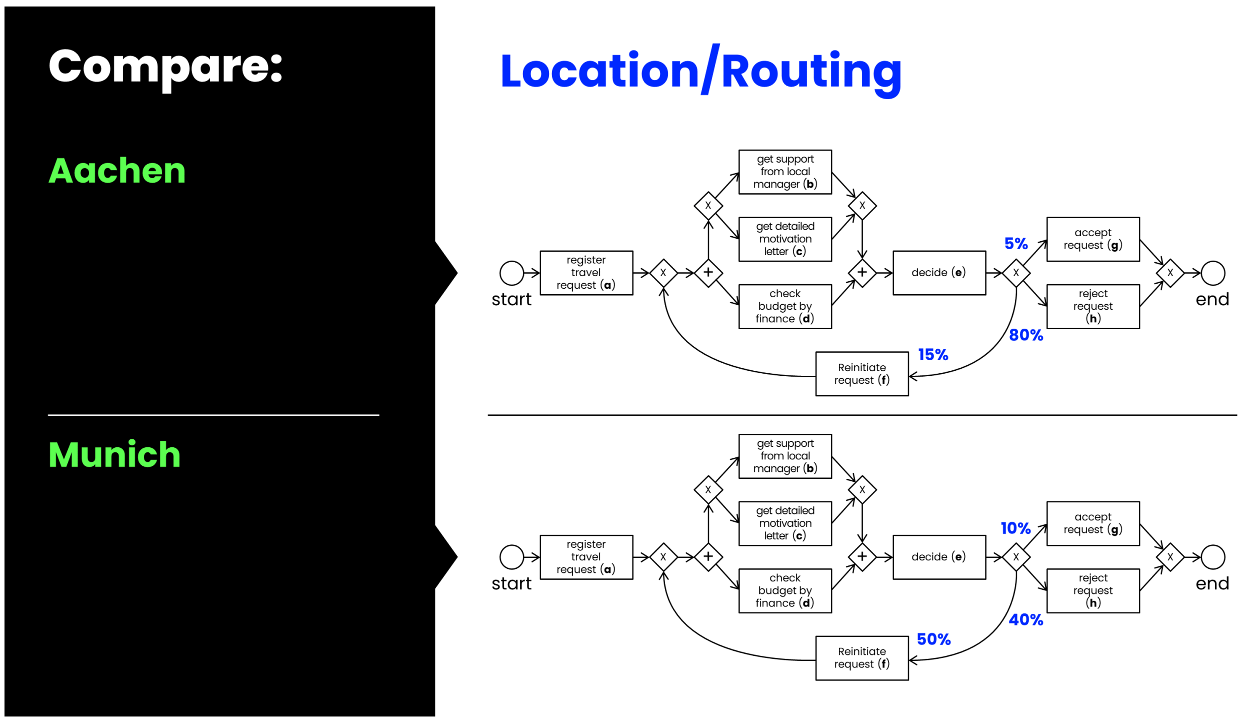
The multi-perspective process model can also be used to generate a simulation model to investigate what-if scenarios. By integrating control flow, data flow, decisions, resources, allocation rules, service times, routing probabilities, arrival processes, etc., one can capture all aspects relevant for simulation. The uptake of object-centric process mining will enable the creation of more realistic simulation models, often referred to as *digital twins*.

▼ Relationship between Comparative and Predictive Process Mining

▼ Comparative Process Mining

Comparative process does not focus on a single event log and process model combination, it focuses on two or more processes represented by event logs or annotated process models. There are two decisions to be made: which processes to compare and which perspectives to compare. This is illustrated using a few examples.

Assume the same processes are performed at two *locations*, and we are interested in *control-flow* differences. By comparing both processes, we may find out that certain paths or activities are more frequent at one location.



Compare:

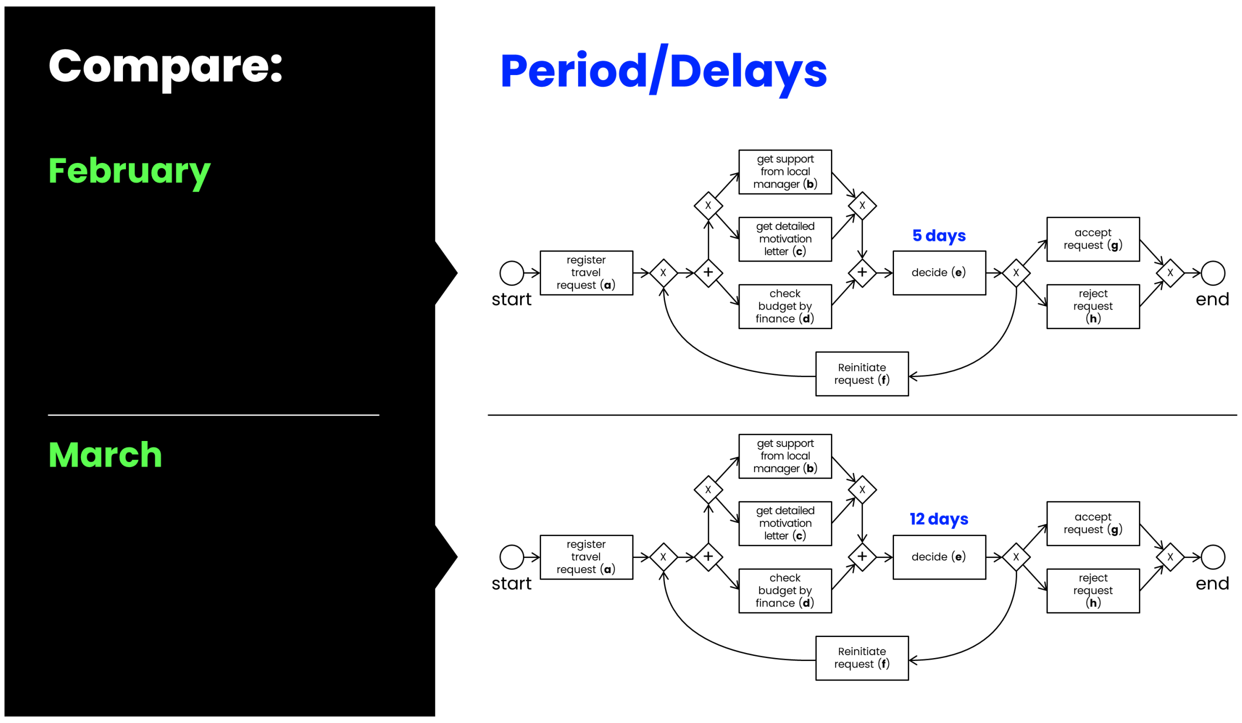
Aachen

Munich

Location/Routing

start
register travel request (a)
get support from local manager (b)
get detailed motivation letter (c)
check budget by finance (d)
decide (e)
Reinitiate request (f)
accept request (g)
reject request (h)
end
5%
80%
15%
10%
40%
50%

We can also compare the same process in *two different periods*. For example, we can compare the cases that started in February with those that started in March. Next to comparing control-flow differences, we may consider differences in time or performance. For example, the bottleneck identified in March did not exist in February.



Compare:

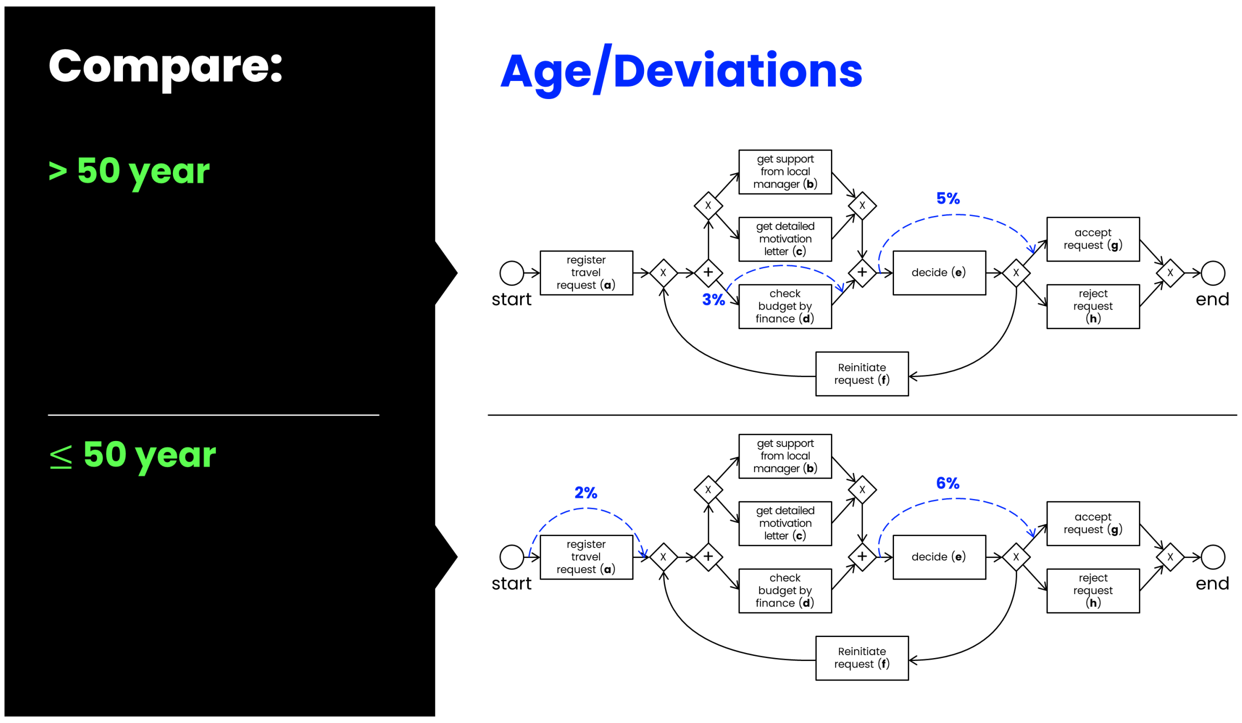
February

March

Period/Delays

start
register travel request (a)
get support from local manager (b)
get detailed motivation letter (c)
check budget by finance (d)
decide (e)
Reinitiate request (f)
accept request (g)
reject request (h)
end
5 days
12 days

We can split cases based on their start times, but also on any other attribute, e.g., age, gender, value, brand, etc. Next, to analyze control-flow and time differences, we can also compare conformance diagnostics.



Compare:

> 50 year

≤ 50 year

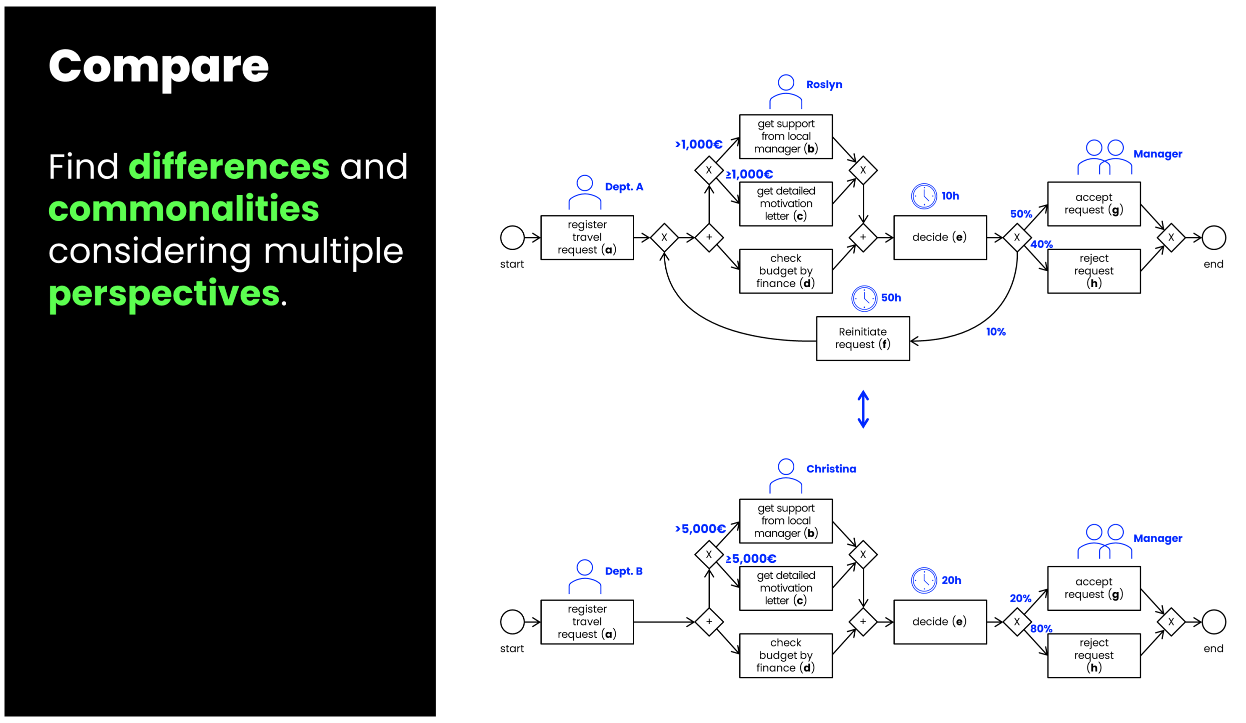
Age/Deviations

start
register travel request (a)
get support from local manager (b)
get detailed motivation letter (c)
check budget by finance (d)
decide (e)
Reinitiate request (f)
accept request (g)
reject request (h)
end
5%
3%
2%
6%

It is also possible to compare processes based on resource utilization, customer satisfaction, costs, etc. It all depends on the event attributes. To facilitate comparison, event data can be organized in a so-called *process cube* where events are organized using different dimensions (e.g., case types, regions, subprocesses, departments, and time windows). The cells in such a process cube can be analyzed using process mining techniques by creating a sub-log per cell. The results of different cells can be compared. It is possible to compare process mining results generated for an array of cells with the goal of highlighting differences between cells.

Comparative process mining is related to *concept-drift* detection. Although the process is already dynamically handling cases, also the process itself may change. Such changes may be unintentional, e.g., waiting times are gradually increasing or the routing probabilities change. There exist techniques to detect such concept drifts, and comparative process mining can be used to highlight the statistically significant process changes.

Assume we would like to compare two processes: **X** and **Y**. These make correspond to different time periods, different products, different locations, etc. Comparative process mining can be based on the events logs of **X** and **Y**, the discovered process models of **X** and **Y**, or both logs and models. There are various techniques to do this.



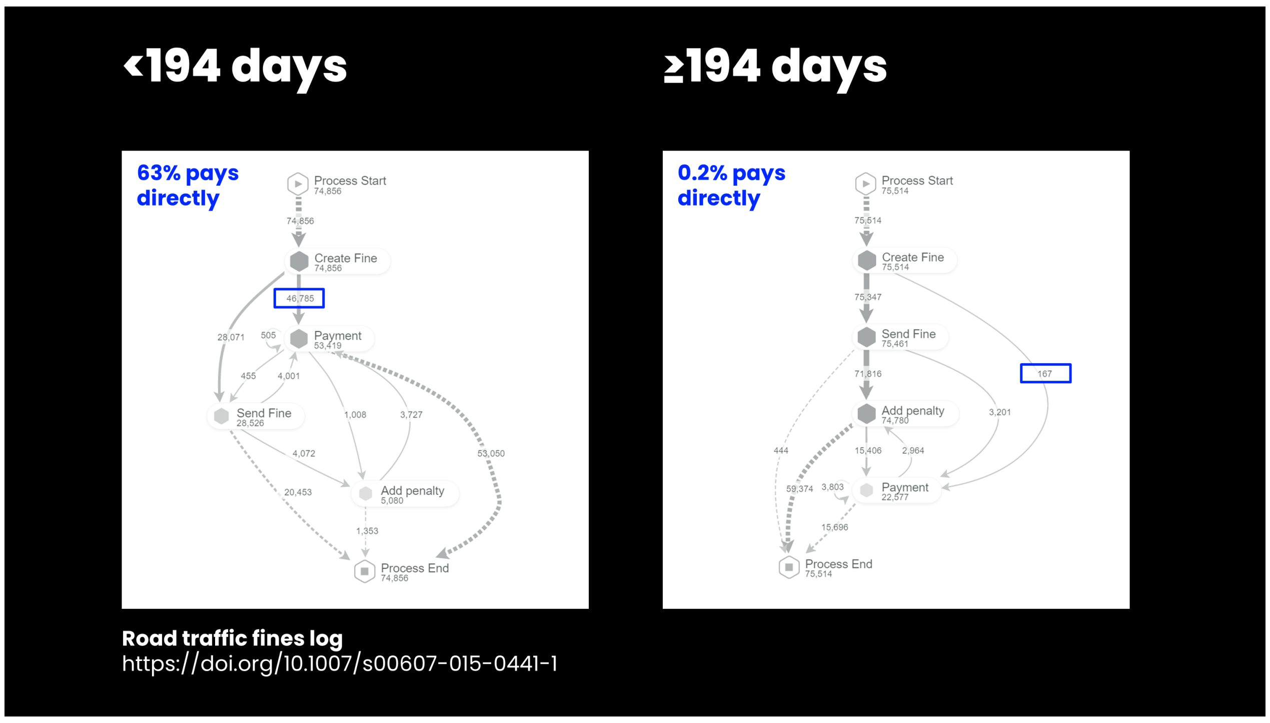
Compare

Find **differences** and **commonalities** considering multiple **perspectives**.

Roslyn
get support from local manager (b)
>1,000€
≥1,000€
Dept. A
register travel request (a)
get detailed motivation letter (c)
check budget by finance (d)
10h
decide (e)
50%
40%
Manager
accept request (g)
reject request (h)
start
end
50h
Reinitiate request (f)
10%
Christina
get support from local manager (b)
>5,000€
≥5,000€
Dept. B
register travel request (a)
get detailed motivation letter (c)
check budget by finance (d)
20h
decide (e)
20%
80%
Manager
accept request (g)
reject request (h)
start
end

Consider, for example, the DFGs of processes **X** and **Y**. By aligning the DFGs, one can immediately see the main differences. Certain connections in the DFG for **X** may be missing in the DFG for **Y**. Even if a connection is present in both DFGs, the frequencies or average flow times may differ significantly. It is also possible to learn or create a process model for **X** and then use the event log of **Y** and the model for **X** for conformance checking. This will show the cases in process **Y** that do not fit into the model for process **X**. In summary, there are many ways to compare two processes using relatively simple methods.

To further illustrate comparative process mining, we take a publically available event log: the so-called *Road traffic fines log* (https://doi.org/10.1007/s00607-015-0441-1). The log contains information about more than 140,000 road-traffic fines with events relating to notifications, payments, and appeals. The cases in the log are split into *short-running* cases (less than 194 days) and *long-running* cases (at least 194 days).

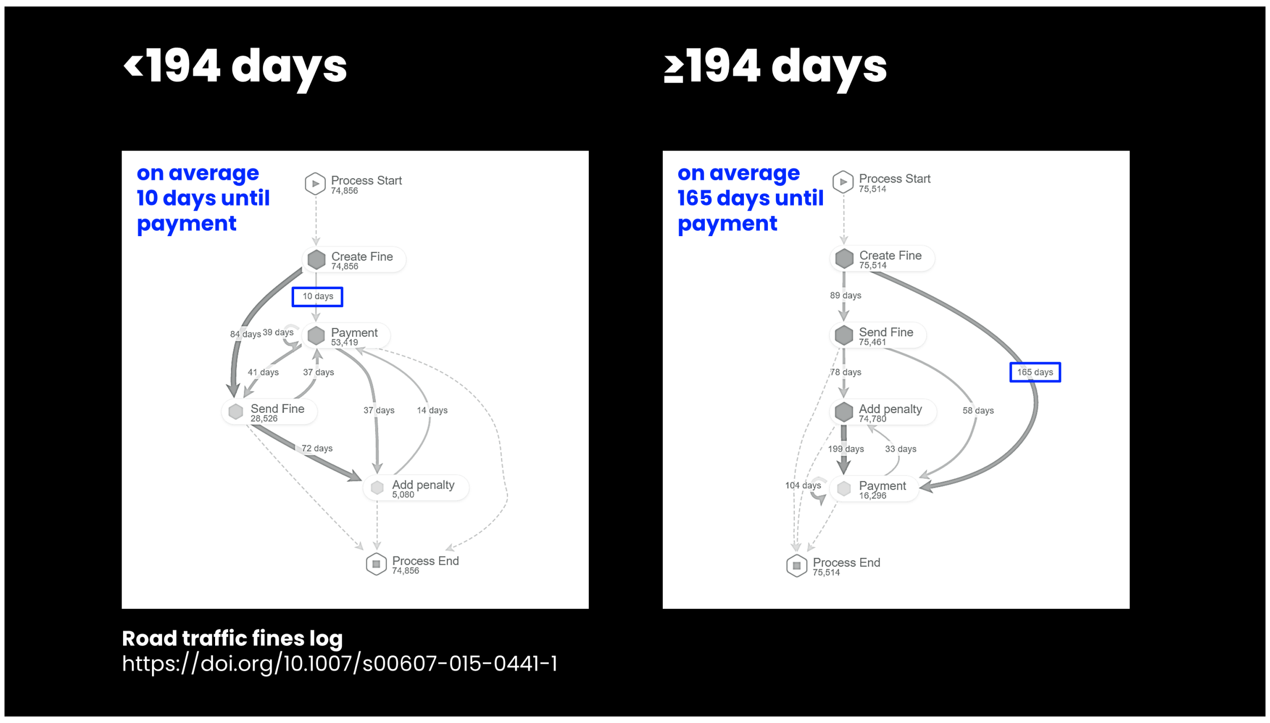


<194 days

≥194 days

Road traffic fines log
https://doi.org/10.1007/s00607-015-0441-1

For the short-running cases (less than 194 days), 63% percent pay immediately. For the long-running cases (at least 194 days), less than 1% pays directly.



‹194 days

≥194 days

on average 10 days until payment

on average 165 days until payment

Road traffic fines log
https://doi.org/10.1007/s00607-015-0441-1

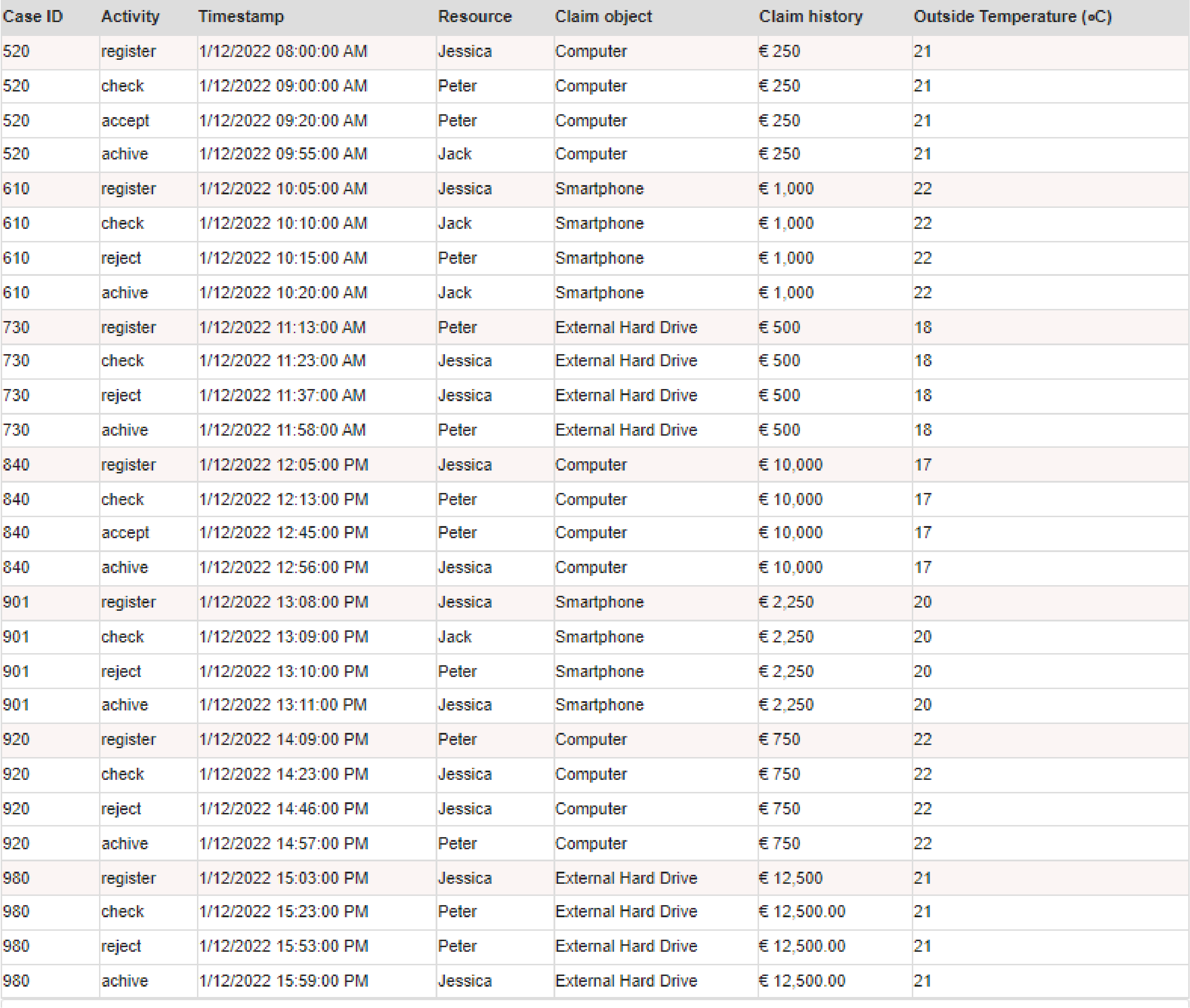
For the short-running cases, payment follows in 10 days on average. For the long-running cases, this is over 100 days. These differences are obvious, but show that it is helpful to compare processes.

▼ Exercice

Question 1

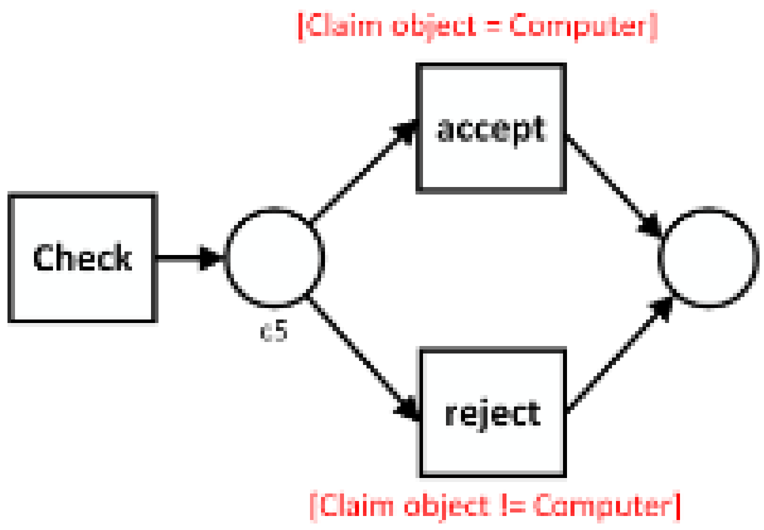
1/1 point (ungraded)

Given the event log below, which of the Petri net parts that are enriched with red guards, fits this data perfectly (i.e. ignoring the activities register and archive)?

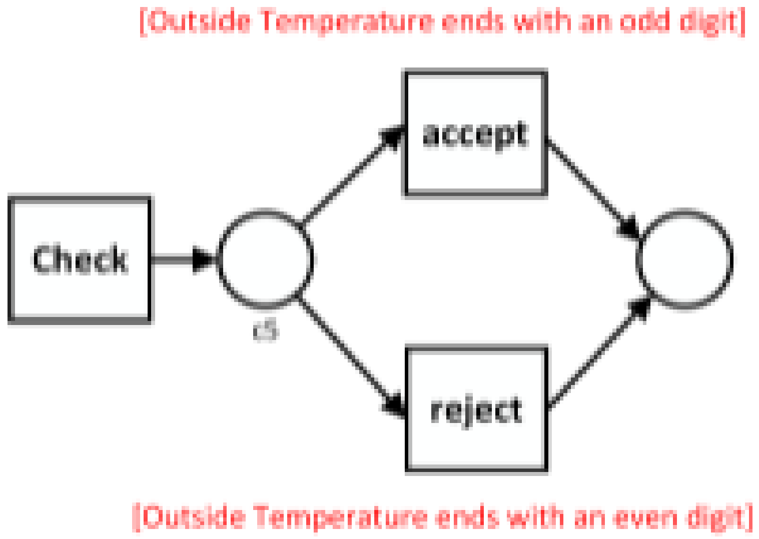


Case ID	Activity	Timestamp	Resource	Claim object	Claim history	Outside Temperature (∘C)
520	register	1/12/2022 08:00:00 AM	Jessica	Computer	€ 250	21
520	check	1/12/2022 09:00:00 AM	Peter	Computer	€ 250	21
520	accept	1/12/2022 09:20:00 AM	Peter	Computer	€ 250	21
520	achive	1/12/2022 09:55:00 AM	Jack	Computer	€ 250	21
610	register	1/12/2022 10:05:00 AM	Jessica	Smartphone	€ 1,000	22
610	check	1/12/2022 10:10:00 AM	Jack	Smartphone	€ 1,000	22
610	reject	1/12/2022 10:15:00 AM	Peter	Smartphone	€ 1,000	22
610	achive	1/12/2022 10:20:00 AM	Jack	Smartphone	€ 1,000	22
730	register	1/12/2022 11:13:00 AM	Peter	External Hard Drive	€ 500	18
730	check	1/12/2022 11:23:00 AM	Jessica	External Hard Drive	€ 500	18
730	reject	1/12/2022 11:37:00 AM	Jessica	External Hard Drive	€ 500	18
730	achive	1/12/2022 11:58:00 AM	Peter	External Hard Drive	€ 500	18
840	register	1/12/2022 12:05:00 PM	Jessica	Computer	€ 10,000	17
840	check	1/12/2022 12:13:00 PM	Peter	Computer	€ 10,000	17
840	accept	1/12/2022 12:45:00 PM	Peter	Computer	€ 10,000	17
840	achive	1/12/2022 12:56:00 PM	Jessica	Computer	€ 10,000	17
901	register	1/12/2022 13:08:00 PM	Jessica	Smartphone	€ 2,250	20
901	check	1/12/2022 13:09:00 PM	Jack	Smartphone	€ 2,250	20
901	reject	1/12/2022 13:10:00 PM	Peter	Smartphone	€ 2,250	20
901	achive	1/12/2022 13:11:00 PM	Jessica	Smartphone	€ 2,250	20
920	register	1/12/2022 14:09:00 PM	Peter	Computer	€ 750	22
920	check	1/12/2022 14:23:00 PM	Jessica	Computer	€ 750	22
920	reject	1/12/2022 14:46:00 PM	Jessica	Computer	€ 750	22
920	achive	1/12/2022 14:57:00 PM	Peter	Computer	€ 750	22
980	register	1/12/2022 15:03:00 PM	Jessica	External Hard Drive	€ 12,500	21
980	check	1/12/2022 15:23:00 PM	Peter	External Hard Drive	€ 12,500.00	21
980	reject	1/12/2022 15:53:00 PM	Peter	External Hard Drive	€ 12,500.00	21
980	achive	1/12/2022 15:59:00 PM	Jessica	External Hard Drive	€ 12,500.00	21

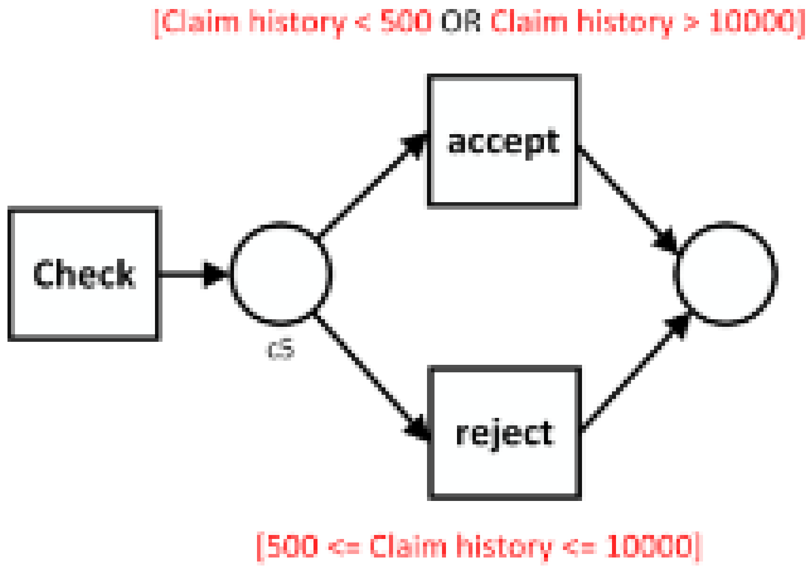
○


[Claim object = Computer]
accept
Check
c5
reject
[Claim object != Computer]

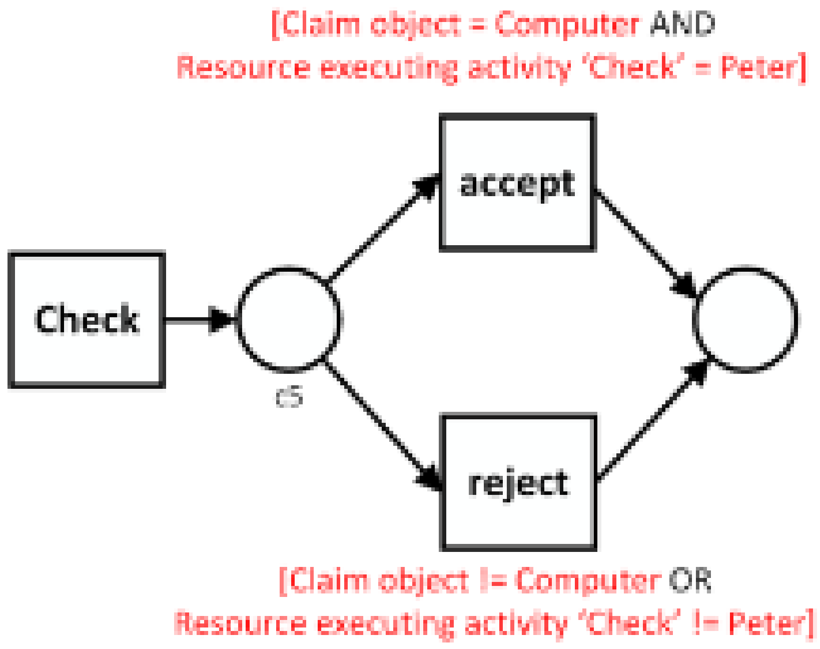
○


[Outside Temperature ends with an odd digit]
accept
Check
c5
reject
[Outside Temperature ends with an even digit]

○


[Claim history < 500 OR Claim history > 10000]
accept
Check
c5
reject
[500 <= Claim history <= 10000]

●

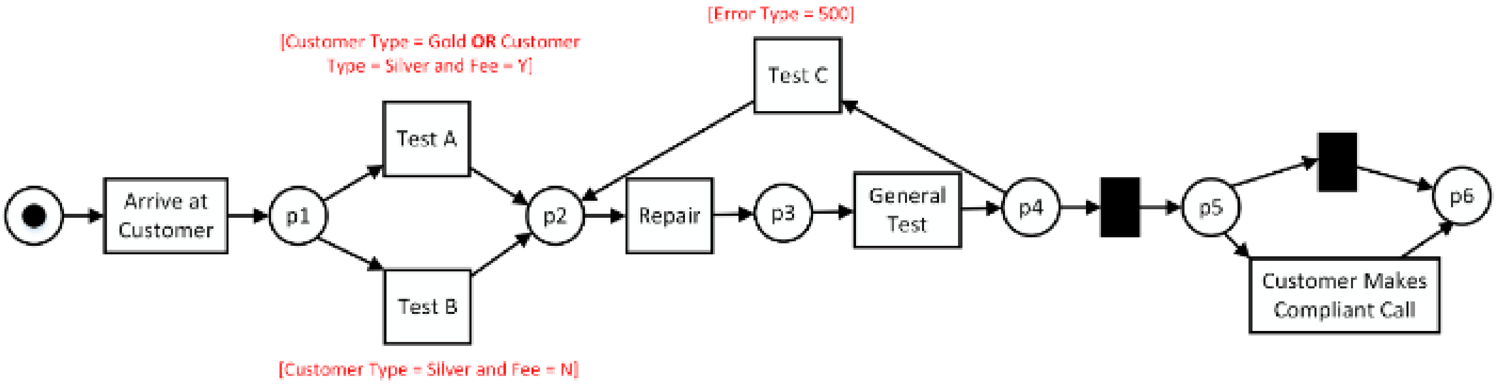

[Claim object = Computer AND
Resource executing activity 'Check' = Peter]
accept
Check
c5
reject
[Claim object != Computer OR
Resource executing activity 'Check' != Peter]

✓

Question 2

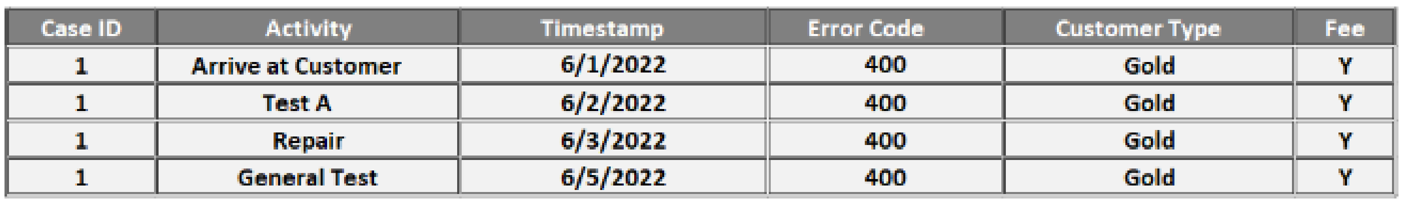
1/1 point (ungraded)

Which trace(s) **cannot** be replayed by the following Petri net model with guards?



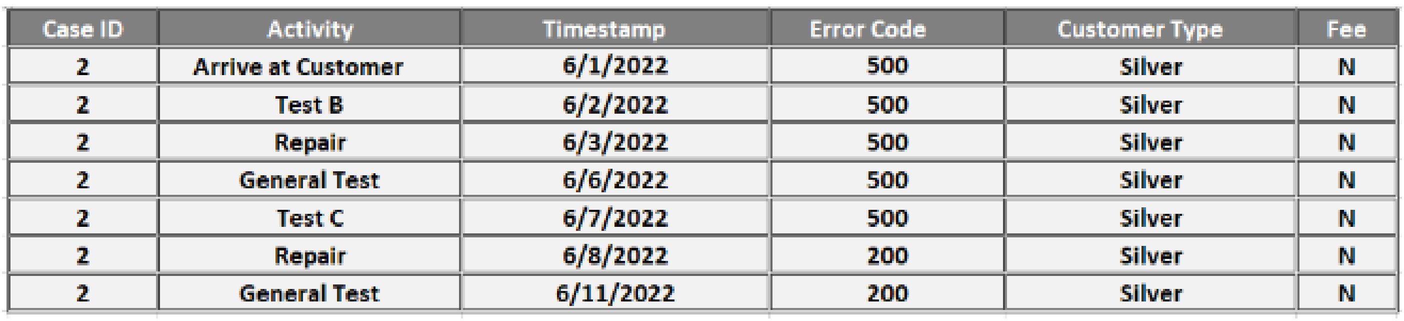
[Error Type = 500]
[Customer Type = Gold **OR** Customer Type = Silver and Fee = Y]
Test C
Test A
Arrive at Customer
p1
p2
Repair
p3
General Test
p4
p5
p6
Customer Makes Compliant Call
Test B
[Customer Type = Silver and Fee = N]

□



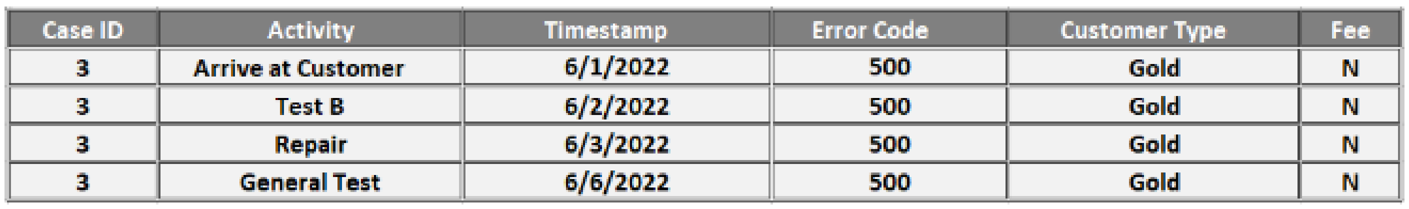
Case ID	Activity	Timestamp	Error Code	Customer Type	Fee
1	Arrive at Customer	6/1/2022	400	Gold	Y
1	Test A	6/2/2022	400	Gold	Y
1	Repair	6/3/2022	400	Gold	Y
1	General Test	6/5/2022	400	Gold	Y

□



Case ID	Activity	Timestamp	Error Code	Customer Type	Fee
2	Arrive at Customer	6/1/2022	500	Silver	N
2	Test B	6/2/2022	500	Silver	N
2	Repair	6/3/2022	500	Silver	N
2	General Test	6/6/2022	500	Silver	N
2	Test C	6/7/2022	500	Silver	N
2	Repair	6/8/2022	200	Silver	N
2	General Test	6/11/2022	200	Silver	N

☑



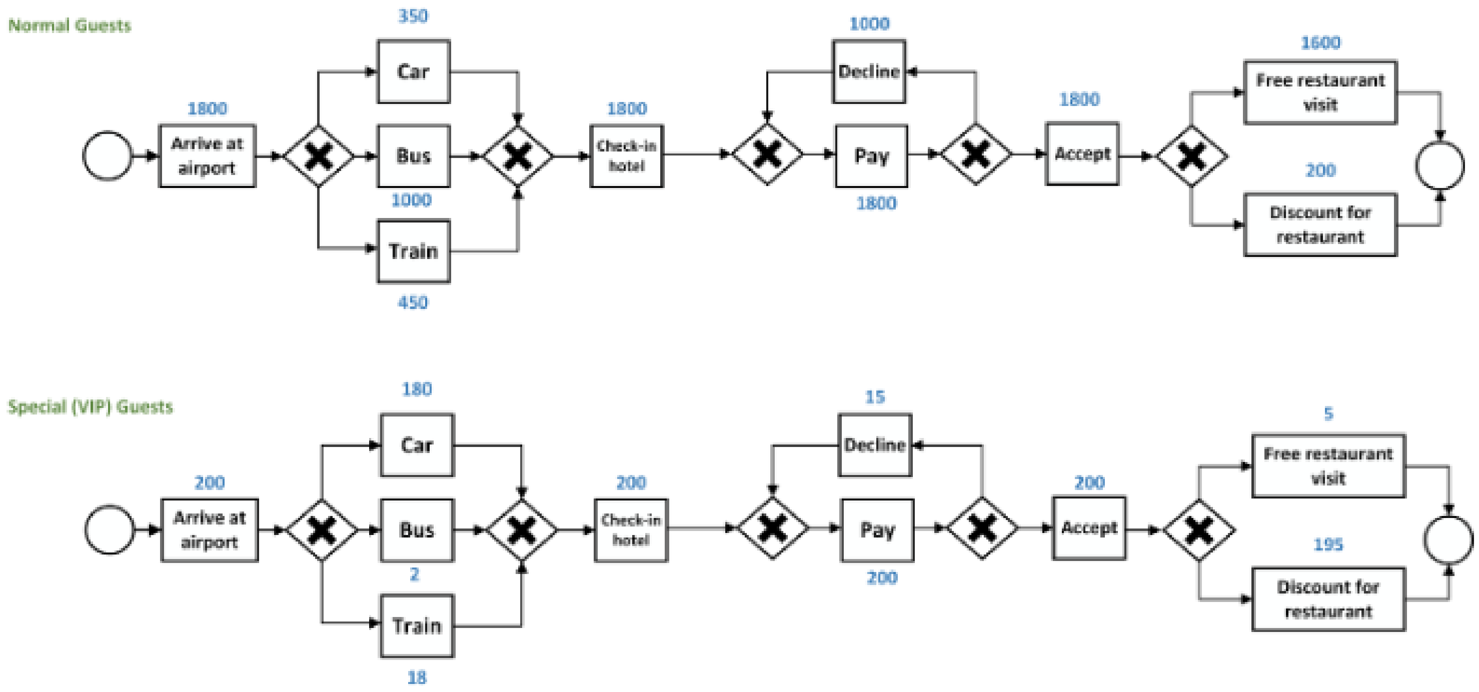
Case ID	Activity	Timestamp	Error Code	Customer Type	Fee
3	Arrive at Customer	6/1/2022	500	Gold	N
3	Test B	6/2/2022	500	Gold	N
3	Repair	6/3/2022	500	Gold	N
3	General Test	6/6/2022	500	Gold	N

✓

Question 3

1/1 point (ungraded)

A hotel records the actions of its guests. The process from the local airport to the welcome dinner is visualized in the BPMN models depicted below for special (VIP) and normal guests. What differences can you spot in behavior? Tick the correct statements. The diagrams are annotated with the number of instances (cases) that include the activity.



☑ In the majority of cases, the payment (Pay activity) of normal guests is declined (Decline activity).

□ In the majority of cases, the payment (Pay activity) of special guests is declined (Decline activity).

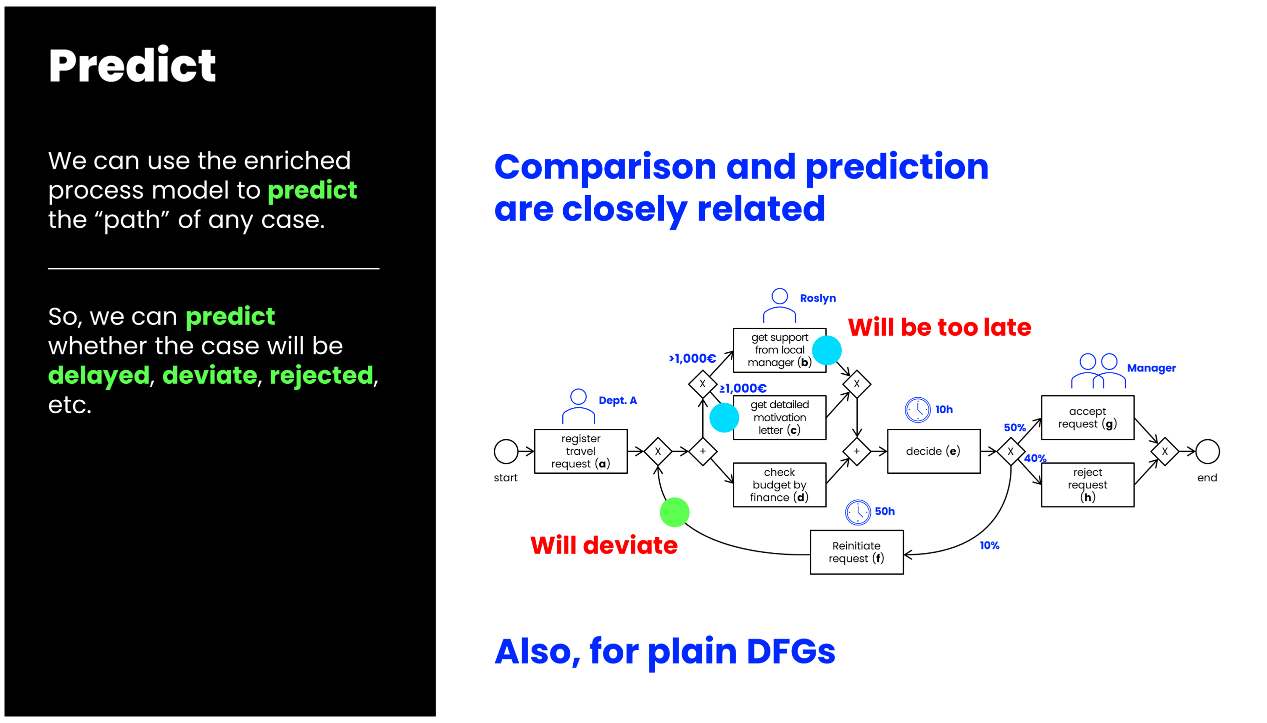
□ Normal guests relatively use cars more than special guests.

☑ Special guests relatively use cars more than normal guests.

✓

▼ Predictive Process Mining

Process discovery and conformance checking are *backward-looking*. The same applies to comparative process mining. Using backward-looking techniques, one can find root causes for performance and compliance problems. This helps to redesign processes and improve control. However, if possible, one would also like to use *forward-looking* techniques. These include predictive techniques based on machine learning and simulation approaches to answer what-if questions.   Forward-looking techniques use models created by the backward-looking approaches discussed before. The enriched process model can for example be used to predict the "path" of a case. This allows us to predict whether a case will be delayed, deviate, succeed, fail, etc.



Predict

We can use the enriched process model to **predict** the "path" of any case.

So, we can **predict** whether the case will be **delayed**, **deviate**, **rejected**, etc.

Comparison and prediction are closely related

Also, for plain DFGs

Consider, for example, the following predictive approach. For a *running case*, we consider the sequence of activities already performed. Using alignments, one can determine the state in the process model where the case is now. Next, one can analyze all previous cases that visited the same state in the process. This way, it is easy to predict the remaining time until completion. Just take the average remaining processing time of all cases in the same state. This approach can be refined to take into account more features of the case and the state of the processes (e.g., workload) and make a more accurate prediction. Next to the remaining processing time, one can predict outcomes or deviations in the same way. It is also possible to use more sophisticated forms of machine learning, but the principle is the same and depends on event data and process mining results.

Examples of predictions at the *case level* are:

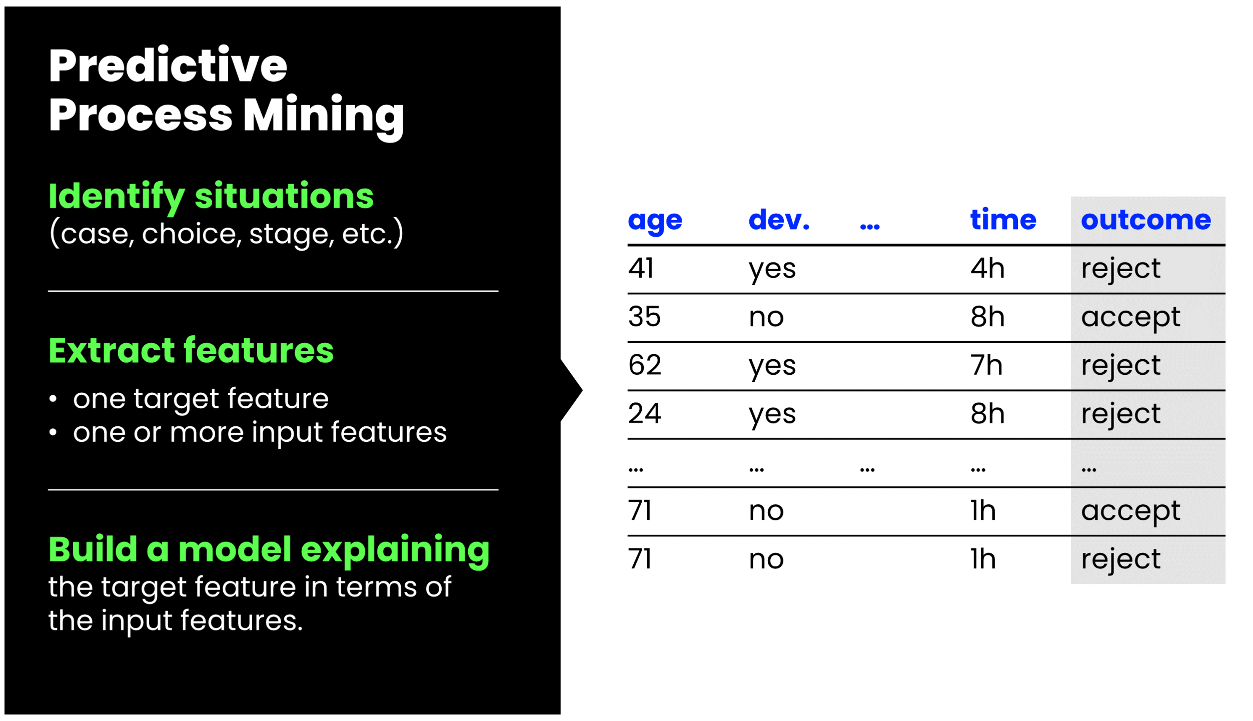
- How long does it take to deliver this delayed order?
- Will this customer churn and switch to a competitor?
- Will this student graduate in time?

Next, to making predictions for running cases, we can also predict *process-level* behavior. For example, what is the likelihood that we can meet a particular Service Level Agreement (SLA). For example, are 80% of the patients with these symptoms admitted to the hospital in 5 days? Such predictions do not relate to a specific case, but say something at an aggregate level. Examples include:

- What will the workload of the Intensive Care Unit (ICU) be tomorrow if we do not intervene?
- How many final products will the production department complete in the coming week?
- How many resources are needed to reduce the backlog?

To predict process-level behavior, simulation is often used. For case-level predictions, typically machine learning is used. All of these techniques rely on process mining to create the models and prepare the data.

As with many data-driven techniques, the challenge is to extract the relevant features from the data in the source systems. The so-called *situation tables* play an essential role in this and provide the interface between process mining and machine learning.



Predictive Process Mining

Identify situations
(case, choice, stage, etc.)

Extract features
- one target feature
- one or more input features

Build a model explaining the target feature in terms of the input features.

age	dev.	...	time	outcome
41	yes		4h	reject
35	no		8h	accept
62	yes		7h	reject
24	yes		8h	reject
...	...	...	...	...
71	no		1h	accept
71	no		1h	reject

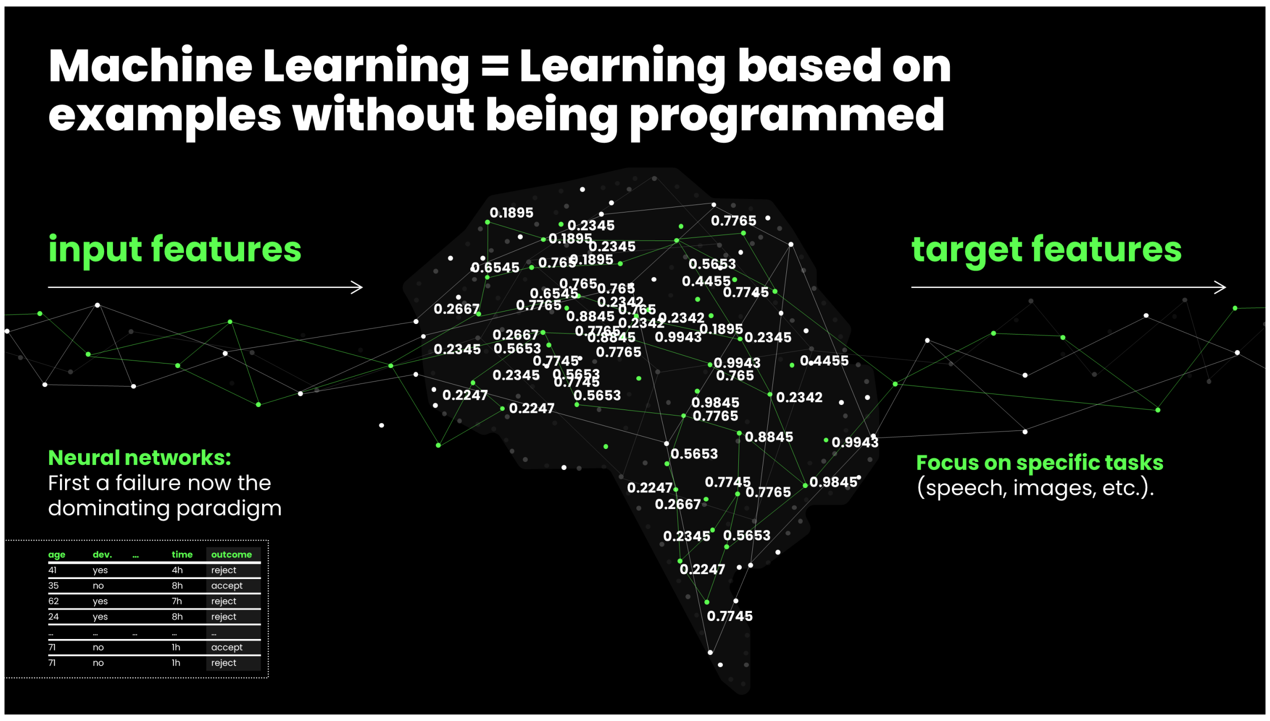
A *situation table* is a two-dimensional table. Each row is an *instance*. Each column is a *feature* (also called attribute or variable). There may be a split into a target feature (also called response variable) and other features (also called predictor variables) if one wants to use supervised learning. However, situation tables are also used for unsupervised learning (e.g., clustering resources or cases). An instance (i.e., row) could be a case, and the target feature could be overall flow time. Other features could be the resources that worked on the case, the number of deviations, and the total number of cases in the pipeline.

We distinguish five types of situation tables:

- *Case-based situation table*: Each row (instance) corresponds to a case with variables.
- *Event-based situation table*: Each row (instance) corresponds to an event.
- *Resource-based situation table*: Each row (instance) corresponds to a resource.
- *Event-pair-based situation table*: Each row (instance) corresponds to a pair of events.
- *Aggregate situation tables*: Each row (instance) corresponds to a combination of cases and/or events.

An event-based situation table could be used to predict a choice in the process (e.g., a decision to accept or reject). Each time a particular choice needs to be made, a new instance is created, just like for decision-point analysis. A resource-based situation table could show how often resources perform activities and use this to cluster resources to find roles. An event-pair-based situation table could be used to analyze the flow time of a subprocess. Each instance would correspond to the first and the last event of the subprocess. The difference between the timestamps of the two events would be the target feature.

The above examples show that a broad range of situation tables can be generated to predict different things. Machine-learning techniques take this as input to train models without being programmed. One can use neural networks, but also more traditional approaches such as decision trees and logistic regression.



Machine Learning = Learning based on examples without being programmed

input features →

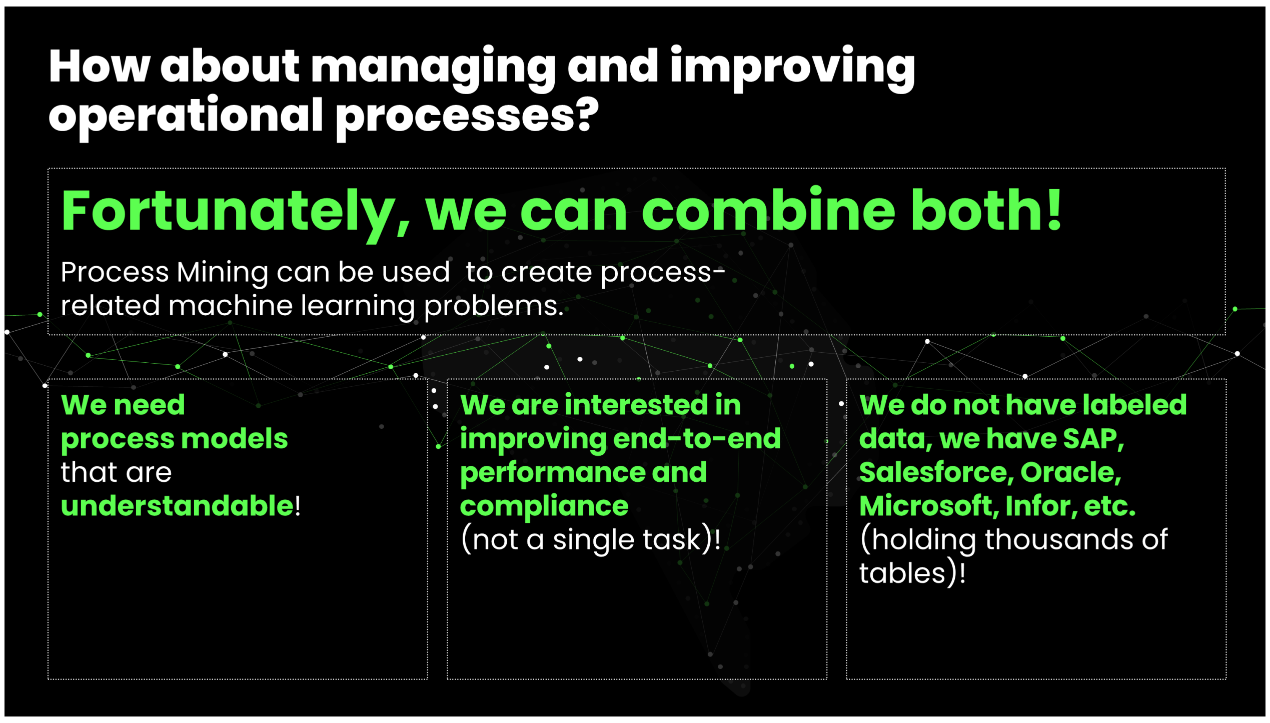
target features →

Neural networks:
First a failure now the dominating paradigm

Focus on specific tasks
(speech, images, etc.).

age	dev.	...	time	outcome
41	yes		4h	reject
35	no		8h	accept
62	yes		7h	reject
24	yes		8h	reject
...	...	...	...	...
71	no		1h	accept
71	no		1h	reject

Process mining focuses on *end-to-end* processes. Mainstream machine-learning techniques cannot directly analyze event data. The focus is often on a particular question. Using situation tables, *process mining is able to provide the input for machine learning*.



How about managing and improving operational processes?

Fortunately, we can combine both!

Process Mining can be used to create process-related machine learning problems.

We need process models that are **understandable**!

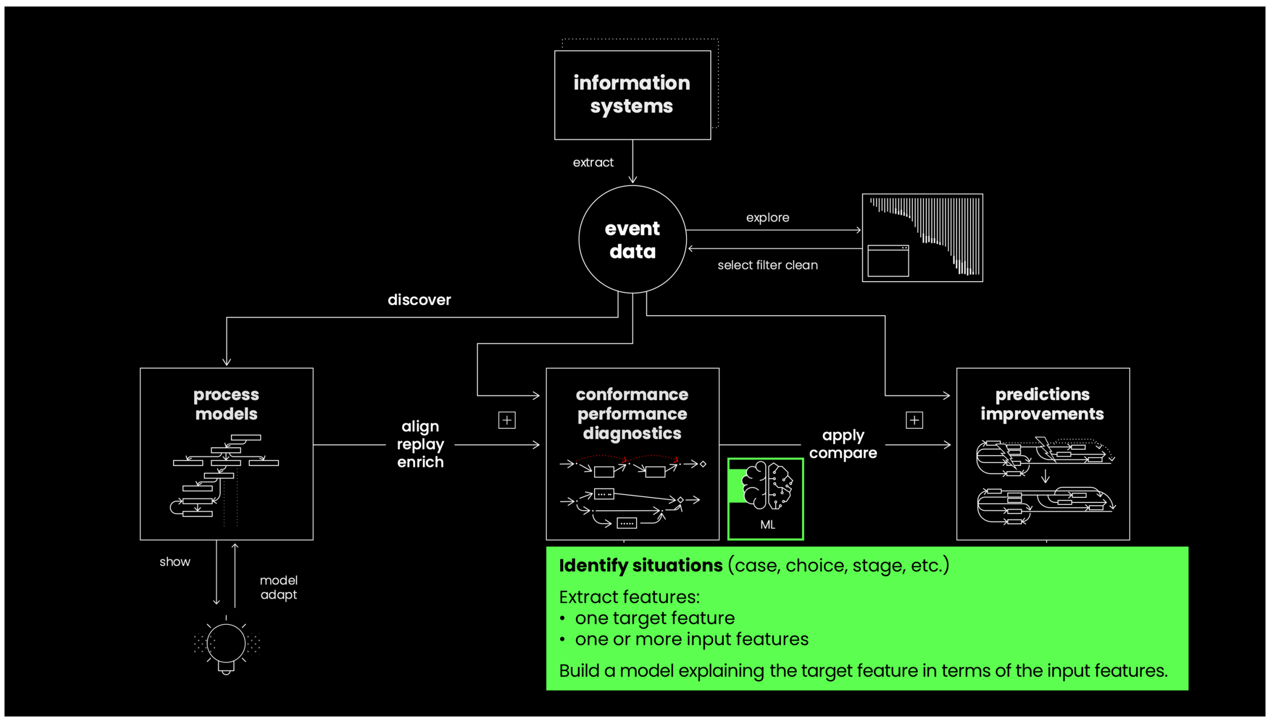
We are interested in improving end-to-end performance and compliance (not a single task)!

We do not have labeled data, we have SAP, Salesforce, Oracle, Microsoft, Infor, etc. (holding thousands of tables)!

Therefore, one can say that process mining *enables* machine learning. However, the core process mining techniques used to discover process models and check compliance are *unrelated* to machine learning. Process mining aims at understandable models, that describe end-to-end processes, and that do not require labeled data. In most cases, the approach is best described as form of *Hybrid Intelligence* (HI). HI combines two forms of intelligence: (1) *human intelligence* is about people and experiences and can be characterized by terms such as flexible, creative, emphatic, instinctive, and commonsensical, and (2) *machine* *intelligence* is about data and algorithms and can be characterized by terms such as fast, efficient, cheap, scalable, and consistent. When it comes to processes, human intelligence can only be exploited if the process models and visualizations are understandable by the stakeholders (i.e., not black-box models like a neural network with millions of neurons).

▼ Before Applying Machine Learning

If we zoom out, it becomes clear that one should start with *backward-looking* techniques like process discovery, conformance checking, performance analysis, and comparative process mining. If there is enough data and the models are of good quality, one can start extracting situation tables and apply machine learning.



Identify situations (case, choice, stage, etc.)

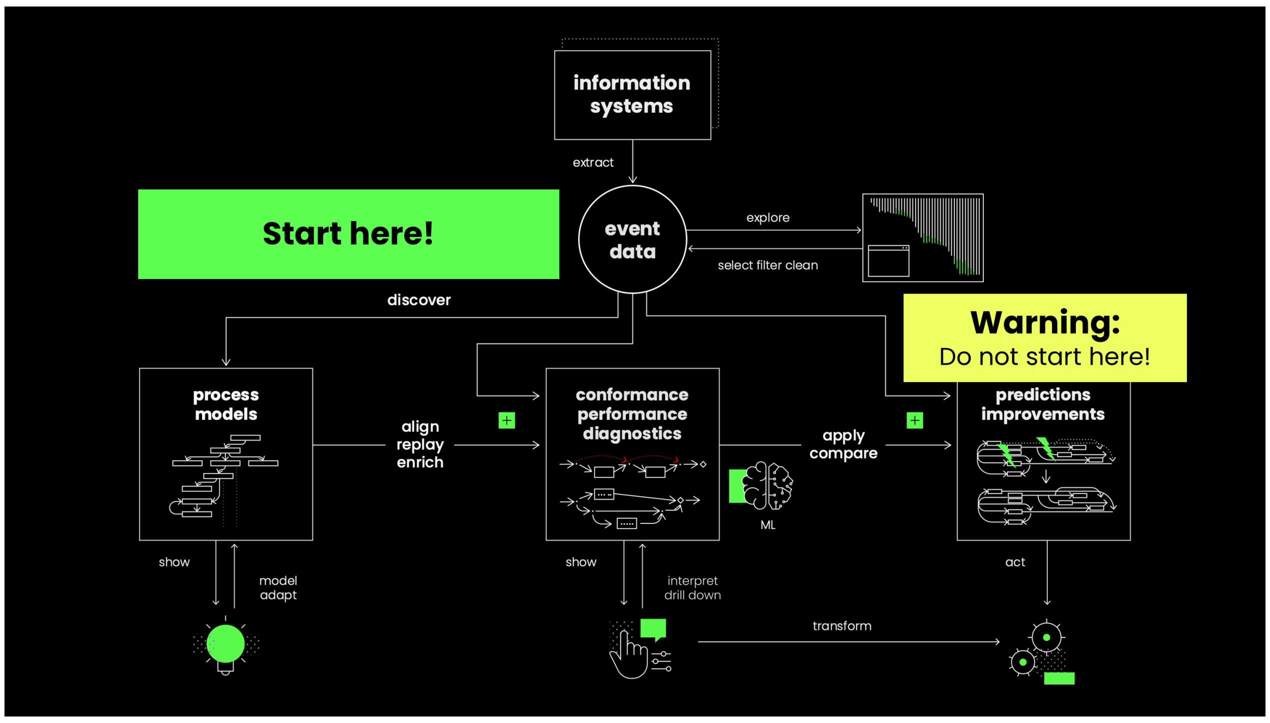
Extract features:

- one target feature
- one or more input features

Build a model explaining the target feature in terms of the input features.

Using the combination of situation tables and machine learning, it is possible to answer questions like:

- How long will this case take?
- Will this case deviate?
- Will this case be rejected?
- What is the next activity?
- Will there be a bottleneck tomorrow?
- Should I accept new cases?
- Should I reallocate people?



However, it is important to start with process discovery, conformance checking, performance analysis, and comparative process mining. If it is impossible to create a proper process model, e.g., because of a lack of data or high variability, then there is no point in starting with machine learning to answer process-related questions. Such attempts are destined to fail.

▼ Celonis - Comparative and Predictive Process Mining

You have reached the final Celonis hands-on session of this course! In this last session, we will see how we can go from discovering potential influencers through comparative process mining to developing direct predictions from then. we will look at an Order Management Use Case in which we want to predict late deliveries. We want to figure out the potential root causes of a delay. This could be specific activities occurring earlier in the process, such as delivery blocks for example. We will use our historical process knowledge from previous data to predict which activities might likely trigger delivery delays.

- https://www.youtube.com/watch?v=-9wXSCJL1fM