Process Mining

By Kourosh Hasani

Kourosh Hasani

BI Manager at Farabi Brokerage

- کارشناسی ارشد مهندسی صنایع از دانشگاه صنعتی شریف
- کارشناسی مهندسی صنایع از دانشگاه صنعتی خواجه نصیر الدین طوسی
- شش سال سابقه فعالیت در حوزه هوشمندسازی کسب‌وکار

- برخی از پروژه‌های انجام شده:
  - Analyzing Customer Journey with Process Mining, from Discovery to Recommendation
  - Develop a web application to analyze the textual content of social networks

1. P.M. Definition
2. P.M. Softwares
3. Advanced P.M. with python
4. Object-Centric P.M.
5. Prediction based on P.M.

1 Process Mining Definition

1.1. What is Process?

Definition:

- A business process or business method is a collection of related, structured activities or tasks that in a specific sequence produces a service or product (serves a particular business goal)

1.1. What is Process?

- If you can't describe what you are doing as a process, you don't know what you're doing

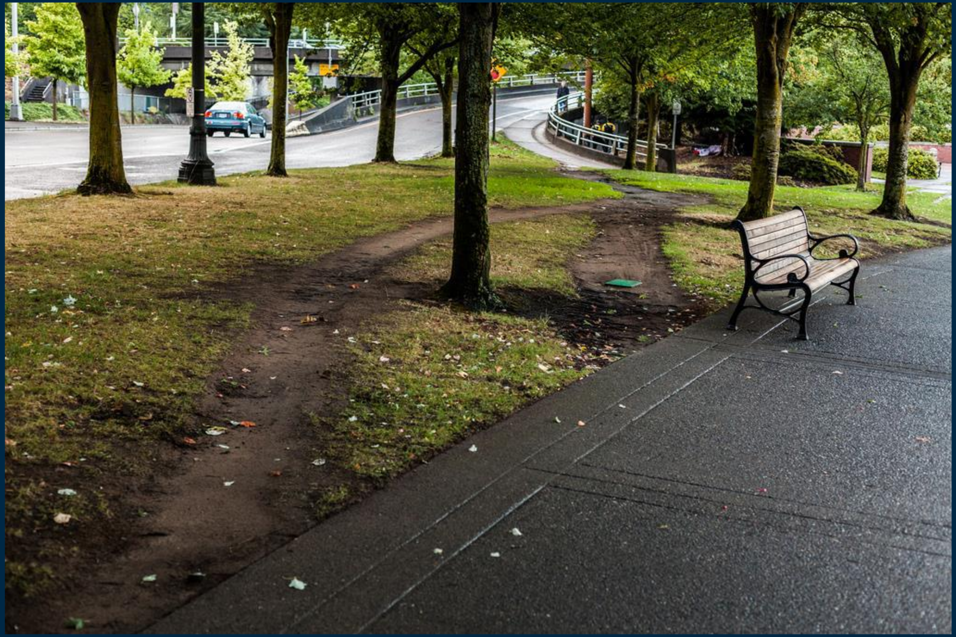
W. Edwards Deming
Leading Management Thinker in the Field of Quality
(1900 – 1993)

1.3. What is Process Mining?

- The idea of *process mining* is to *discover*, *monitor* and *improve* "**real processes**" (not assumed processes) by extracting *knowledge* from "**event-logs**" readily available in todays (information) systems.

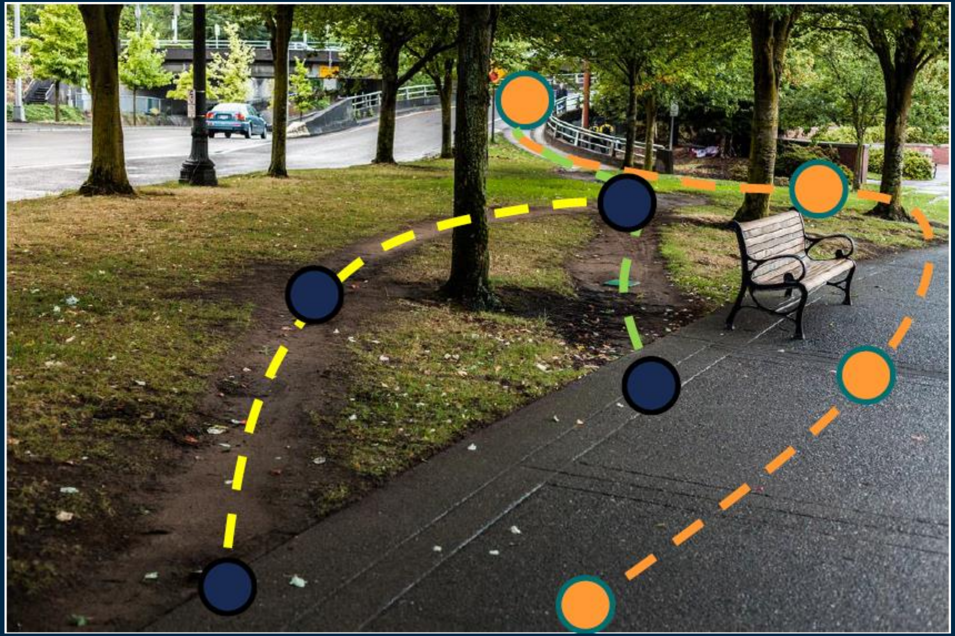
Prof.dr.ir. Wil van der Aalst
Father of Process Mining

1.4. Real VS Assumed Process



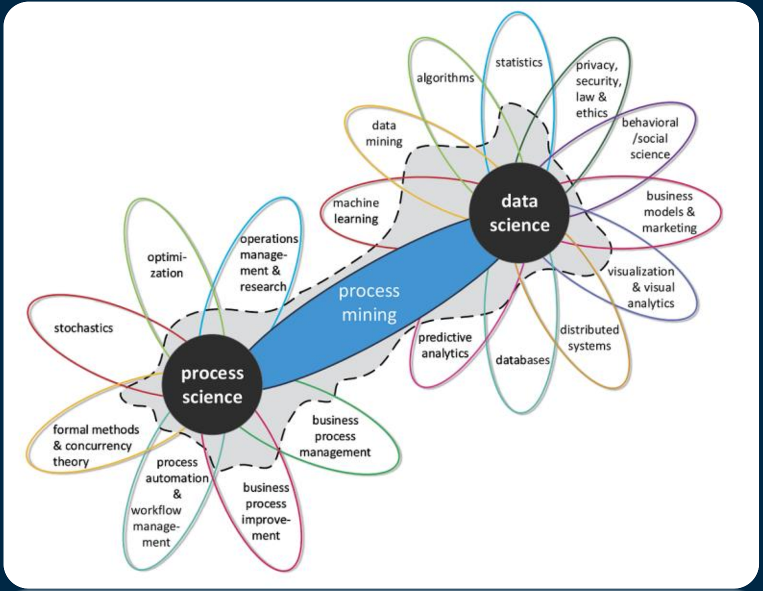
1.4. Real VS Assumed Process

1.4. Real VS Assumed Process



1.5. Process Mining - Data Science

- *Process mining is a type of data analytics that focuses on the discovery, monitoring, and improvement of business processes.*



1.6. Process Mining Applications:

1. Process mining and RPA:

Robotic process automation or RPA is focused on automating repetitive business processes to increase efficiency.

1.6. Process Mining Applications:

2. Process mining and supply chain

1.6. Process Mining Applications:

3. Process mining and finance

Process mining can help in the "Risk management" process.

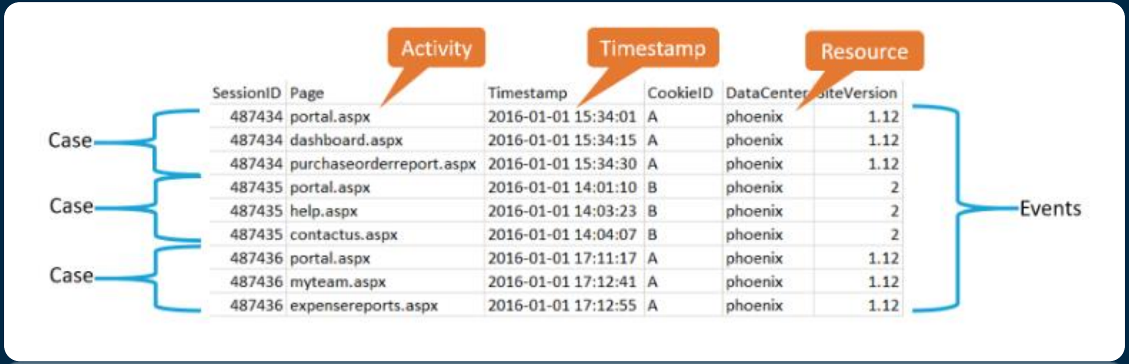
1.7. Event-Log

- An event-log can be seen as a collection of cases and a case can be seen as a trace/sequence of events.

- Attributes that are typically listed in an event-log are:
  - Case ID
  - Activity or Event Name
  - Timestamps of the start and end times
  - Other attributes of the case or event

1.7. Event-Log

- A process consists of CASES
- A case consists of EVENTS
- Events within a case are ORDERED
- Events can have ATTRIBUTES (cost, resource, activity)



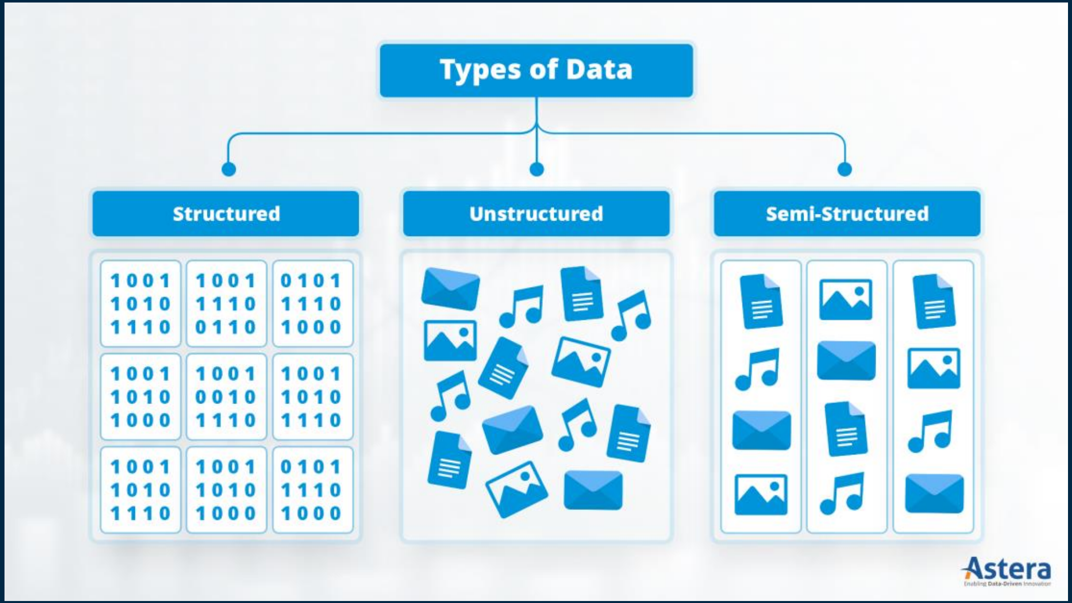
SessionID	Page	Timestamp	CookieID	DataCenter	SiteVersion
487434	portal.aspx	2016-01-01 15:34:01	A	phoenix	1.12
487434	dashboard.aspx	2016-01-01 15:34:15	A	phoenix	1.12
487434	purchaseorderreport.aspx	2016-01-01 15:34:30	A	phoenix	1.12
487435	portal.aspx	2016-01-01 14:01:10	B	phoenix	2
487435	help.aspx	2016-01-01 14:03:23	B	phoenix	2
487435	contactus.aspx	2016-01-01 14:04:07	B	phoenix	2
487436	portal.aspx	2016-01-01 17:11:17	A	phoenix	1.12
487436	myteam.aspx	2016-01-01 17:12:41	A	phoenix	1.12
487436	expensereports.aspx	2016-01-01 17:12:55	A	phoenix	1.12

1.7. Event-Log

Event log example:

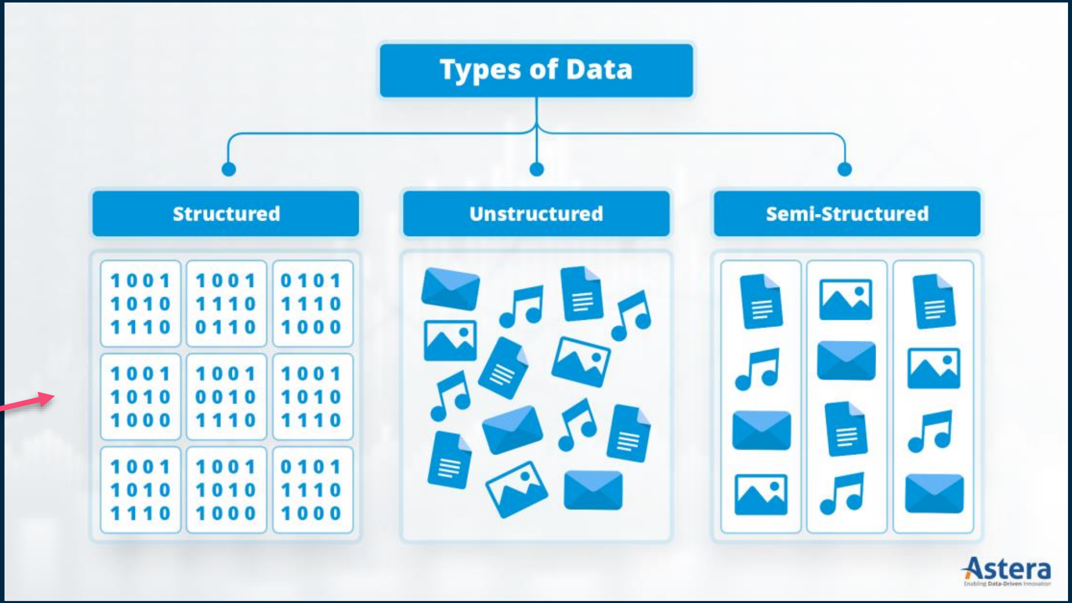
User_ID	CaseStartDate	ActivityStartDate	Browser	OS	Device	Country	Visited_Page	TimeOnPage
hv5xru	12/30/2021 23:58	12/30/2021 23:58	ChromeMobile	Android	Mobile	Iran	learning.emofid.com	168
hv5xru	12/30/2021 23:58	12/31/2021 0:00	ChromeMobile	Android	Mobile	Iran	.../online-issuance-and-cancellation/	153
hv5xru	12/30/2021 23:58	12/31/2021 0:03	ChromeMobile	Android	Mobile	Iran	learning.emofid.com	5
hv5xru	12/30/2021 23:58	12/31/2021 0:04	ChromeMobile	Android	Mobile	Iran	learning.emofid.com	26
92c26h	12/30/2021 23:57	12/30/2021 23:57	Firefox	Windows	PC	Iran	learning.emofid.com	129
92c26h	12/30/2021 23:57	12/30/2021 23:59	Firefox	Windows	PC	Iran	.../bambo/	32
92c26h	12/30/2021 23:57	12/30/2021 23:59	Firefox	Windows	PC	Iran	learning.emofid.com	23
92c26h	12/30/2021 23:57	12/31/2021 0:00	Firefox	Windows	PC	Iran	.../event/rights-issues-strategy-8day/	76

1.7. Event-Log



Types of Data
Structured
Unstructured
Semi-Structured
1001
1010
1110
1001
1110
0110
0101
1110
1000
1001
1010
1000
1001
0010
1110
1001
1010
1110
1001
1010
1110
1001
1010
1000
0101
1110
1000
Astera

1.7. Event-Log



Event Log

1.7. Event-Log

Formats of event log:

- Comma Separated Values (CSV)
- eXtensible Event Stream (XES)
- Object Centric Event Logs (OCEL)

1.7. Event-Log

Formats of event log:

- Comma Separated Values (CSV)
- eXtensible Event Stream (XES)
- Object Centric Event Logs (OCEL)

Case_ID	Activity	Timestamp
OR-E000002	ارسال درخواست	3/30/2019 10:26
OR-E000002	بررسی سفارش توسط انبار	3/30/2019 10:26
OR-E000002	بررسی مدیر انبار	3/31/2019 12:15
OR-E000002	تأیید معاون درخواست کننده	4/1/2019 14:49
OR-E000002	بررسی بودجه تعدادی توسط سرپرست بازرگانی	4/3/2019 8:44
OR-E000002	کنترل ثبت سفارش داخلی	4/7/2019 19:33
OR-E000002	اقدام کارشناس بازرگانی داخلی برای خرید سفارش	4/8/2019 14:51
OR-E000003	ارسال درخواست	3/30/2019 14:02
OR-E000003	تایید سرپرست یا مدیر	3/30/2019 14:03
OR-E000003	بررسی سفارش توسط انبار	3/31/2019 14:55
OR-E000003	بررسی مدیر انبار	4/1/2019 13:41
OR-E000003	تأیید معاون درخواست کننده	4/1/2019 14:53
OR-E000003	بررسی بودجه تعدادی توسط سرپرست بازرگانی	4/6/2019 8:21
OR-E000003	بررسی بودجه تعدادی توسط سرپرست بازرگانی	4/6/2019 13:31
OR-E000003	تأیید مدیر ارشد بازرگانی	4/6/2019 15:17
OR-E000003	اقدام کارشناس بازرگانی داخلی برای خرید سفارش	4/8/2019 12:49

1.7. Event-Log

Formats of event log:

- *Comma Separated Values (CSV)*
- *eXtensible Event Stream (XES)*
- *Object Centric Event Logs (OCEL)*


```xml
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7"
xmlns="http://www.xes-standard.org/">
    <trace>
        <date key="REG_DATE" value="2011-10-01T00:38:44.546+02:00"/>
        <string key="concept:name" value="173688"/>
        <string key="AMOUNT_REQ" value="20000"/>
        <event>
            <string key="org:resource" value="112"/>
            <string key="lifecycle:transition" value="COMPLETE"/>
            <string key="concept:name" value="A_SUBMITTED"/>
            <date key="time:timestamp" value="2011-10-01T00:38:44.546+02:00"/>
        </event>
        <event>
            <string key="org:resource" value="112"/>
            <string key="lifecycle:transition" value="COMPLETE"/>
            <string key="concept:name" value="A_PARTLYSUBMITTED"/>
            <date key="time:timestamp" value="2011-10-01T00:38:44.880+02:00"/>
        </event>
        <event>
            <string key="org:resource" value="112"/>
            <string key="lifecycle:transition" value="COMPLETE"/>
            <string key="concept:name" value="A_PREACCEPTED"/>
            <date key="time:timestamp" value="2011-10-01T00:39:37.906+02:00"/>
        </event>
        <event>
            <string key="org:resource" value="10862"/>
            <string key="lifecycle:transition" value="COMPLETE"/>
            <string key="concept:name" value="A_ACCEPTED"/>
            <date key="time:timestamp" value="2011-10-01T11:42:43.308+02:00"/>
        </event>
```

1.7. Event-Log

Formats of event log:

- Comma Separated Values (CSV)
- eXtensible Event Stream (XES)
- Object Centric Event Logs (OCEL)

https://ocel-standard.org/


```
{
  "ocel:global-log": {
    "ocel:attribute-names": [
      "age",
      "bankaccount",
      "cost",
      "price",
      "weight"
    ],
    "ocel:version": "1.0",
    "ocel:ordering": "timestamp"
  },
  "ocel:events": {
    "1.0": {
      "ocel:activity": "place order",
      "ocel:timestamp": "2019-05-20T09:07:47",
      "ocel:omap": [
        "880001",
        "Echo Studio",
        "Marco Pegoraro",
        "990001",
        "Fire Stick 4K",
        "Echo"
      ],
      "ocel:vmap": {
        "weight": 3.52,
        "price": 524.96
      }
    },
```

1.8. Alpha algorithm

Steps:

1. Define all events
2. Define all possible start events
3. Define all possible End events
4. Calculate possible Sets A and B (All events within A and within B should be independent of each other. All events in A should be causally related to event in B)
5. Drop redundant sets
6. Draw the Petri Net

Source: Van der Aalst, W., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. IEEE transactions on knowledge and data engineering, 16(9), 1128-1142.

1.8. Alpha algorithm

Order relations:

- *Direct successor:* $a > b$
- *Causality:* $a \rightarrow b$
- *Concurrency:* $a \parallel b$
- *Exclusiveness:* $a \# b$

Source: Van der Aalst, W., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. IEEE transactions on knowledge and data engineering, 16(9), 1128-1142.

1.8. Alpha algorithm

Example:

Event log: [(a, b, c, d) ^3 , (a, c, b, d) ^2 , (a, e, d)]

1. Define all events: (a, b, c, d, e)

2-3. Define all possible start events: (a) - End events: (d)

4. Calculate possible Sets:

[({a}, {b}), ({a}, {c}), ({a}, {e}), ({b}, {d}), ({c}, {d}), ({e}, {d})]

5. Drop redundant sets:

[({a}, {b, e}), ({a}, {c, e}), ({b ,e}, {d}), ({c, e}, {d})]

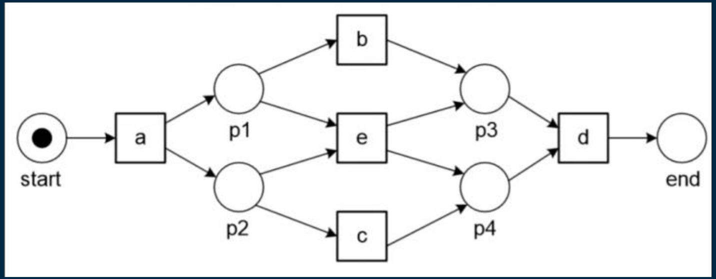
Source: Van der Aalst, W., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. IEEE transactions on knowledge and data engineering, 16(9), 1128-1142.

1.8. Alpha algorithm

Example:

Event log: [(a, b, c, d) ^3 , (a, c, b, d) ^2 , (a, e, d)]

6. Draw the Petri Net:



Source: Van der Aalst, W., Weijters, T., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. IEEE transactions on knowledge and data engineering, 16(9), 1128-1142.

1.9. P.D. Algorithms:

- Alpha algorithm: This is the simplest algorithm. It constructs the Graph by extracting directly follows dependencies from the event log.
- Heuristic Miner: This algorithm also constructs the Graph but then applies heuristics to filter out infrequent or noisy behavior. It results in more structured models.
- Inductive Miner: This algorithm first constructs the Graph and then reduces it by detecting long-range dependencies and looping structures. It results in more structured models than the Heuristic Miner.
- Genetic Miner: This applies genetic algorithms to mine process models that are optimized for a specific criterion, like fitness or simplicity.
- Region-based Miner: This discovers local models (regions) and then combines them into an overall model. It works well for "spaghetti-like" processes with a lot of divergent paths.
- Fodina: This is based on the Region-based Miner approach but mines regions in a bottom-up fashion instead of top-down.
- Heuristics Net Miner: This algorithm mines both control-flow and organizational perspectives and produces Heuristics Net models.

2 Process Mining Softwares

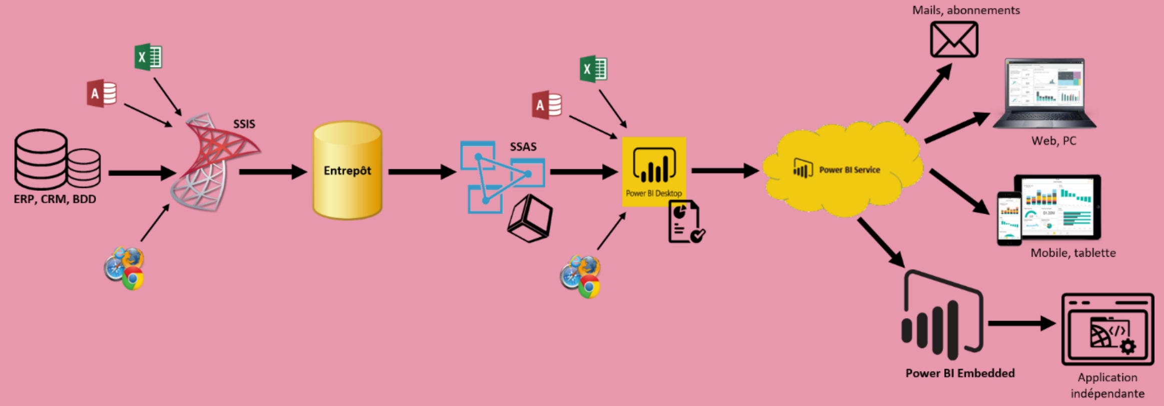
2.1. Process Mining Softwares

The following software will be reviewed:

- Celonis
- Disco
- PmBI and PafNow (in Power-BI)

Figure 1: Magic Quadrant for Process Mining Tools

Process Mining with Power-BI



Why P.M. in Power-BI?

- *Process control and monitoring*
- *Familiarity of organizations with Power-BI*
- *Create user friendly reports*

Process Mining with Power-BI

- *Using ready-made visuals*
- *Write python script*

Process Mining with Power-BI

- *Using ready-made visuals*
- *Write python script*

3

Advanced Process Mining with python

Introduction to Python

- Python is a popular programming language that is widely used in data science due to its simplicity and flexibility.
- Python is an open-source language and has a large community of developers.
- Python can be used to perform a wide range of tasks, from simple data processing to complex machine learning algorithms.
- Python can be used to collect, analyze, and visualize event logs, as well as to build predictive models to forecast process behavior.

Why Python is useful for process mining?

- it offers a wide range of libraries and tools that can be used to perform various tasks in the process mining workflow.
- Some popular libraries for process mining in Python include:
  - pm4py
  - ProM
- With Python, it is also easy to integrate different tools and technologies within the process mining workflow, making it a flexible and versatile choice for process mining projects.

1. Use Python for Performance Analysis

- Python can be used to calculate different performance metrics like cycle time, throughput, and efficiency.
- Python can also be used to visualize performance metrics in different ways, making it easy to identify bottlenecks and areas for improvement in the process.

2. Use Python for Process Discovery

- Python can be used for process discovery by using libraries like pm4py to apply various process discovery algorithms on event log.
- Python libraries like pm4py also offer various process visualization and analysis techniques, making it easy to gain insights into process behavior and performance.

3. Use Python for Conformance Checking

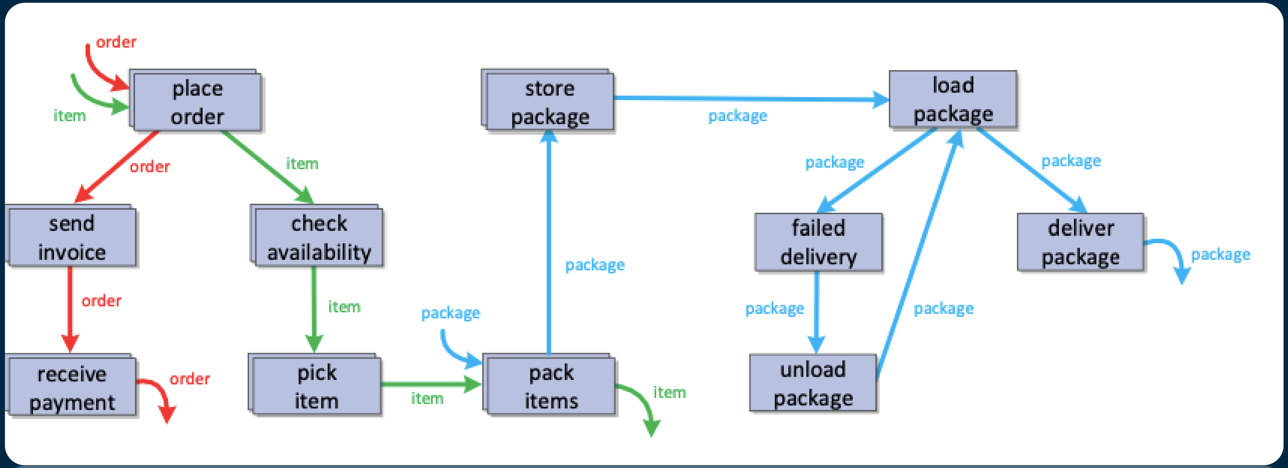
- Python can be used for conformance checking to compare process models with actual event logs.
- Pm4py offer various conformance checking techniques that can be used to identify deviations between the discovered process model and the actual process as captured in the event log.

4. Use Python for Process Improvement

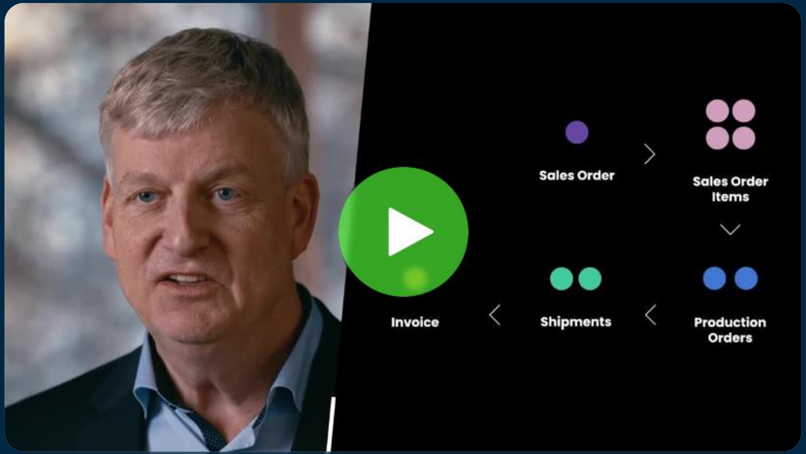
- Python can be a powerful tool for process improvement.
- Because it allows for the analysis of large amounts of event data and the simulation of different scenarios to identify the most effective changes.

4 Object-Centric Process Mining

Object-Centric Process Mining



Object-Centric Process Mining



https://www.celonis.com/blog/what-is-object-centric-process-mining-ocpm/

5

Prediction based on Process Mining

Predictive Process Mining

- Predictive process mining work leverages machine learning and deep learning
- Applications:
  - Mitigating risk: (measure the delays or remaining time)
  - Predicting costs
  - Generating recommendations

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Time prediction based on process mining

W.M.P. van der Aalst [a,*], M.H. Schonenberg [a], M. Song [a,b]

[a] Eindhoven University of Technology, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands
[b] Ulsan National Institute of Science and Technology, 689-798 100 Banyeon-ri, Ulsan, South Korea

ARTICLE INFO

Article history:
Received 4 August 2009
Received in revised form
7 August 2010
Accepted 2 September 2010
Recommended by: M. Weske

Keywords:
Process mining
Business process management
Time prediction
Performance analysis
Business intelligence

ABSTRACT

Process mining allows for the automated discovery of process models from event logs. These models provide insights and enable various types of model-based analysis. This paper demonstrates that the discovered process models can be extended with information to *predict the completion time* of running instances. There are many scenarios where it is useful to have reliable time predictions. For example, when a customer phones her insurance company for information about her insurance claim, she can be given an estimate for the remaining processing time. In order to do this, we provide a configurable approach to construct a process model, augment this model with time information learned from earlier instances, and use this to predict e.g., the completion time. To provide meaningful time predictions we use a configurable set of abstractions that allow for a good balance between "overfitting" and "underfitting". The approach has been implemented in ProM and through several experiments using real-life event logs we demonstrate its applicability.

Predictive Process Mining

1	$\langle A^{00}, B^{06}, C^{12}, D^{18} \rangle$
2	$\langle A^{10}, C^{14}, B^{26}, D^{36} \rangle$
3	$\langle A^{12}, E^{22}, D^{56} \rangle$
4	$\langle A^{15}, B^{19}, C^{22}, D^{28} \rangle$
5	$\langle A^{18}, B^{22}, C^{26}, D^{32} \rangle$
6	$\langle A^{19}, E^{28}, D^{59} \rangle$
7	$\langle A^{20}, C^{25}, B^{36}, D^{44} \rangle$

Each line corresponds to a trace represented as a sequence of activities with timestamps.

- Each full trace is split into a:
  - Prefix: The part that already took place
  - Postfix: The part that still needs to happen

Predictive Process Mining

$< A^{00}, B^{06}, C^{12}, D^{18} >$ refers to a traces.

- State 1: Prefix = $< >$ ---- Postfix = $< A^{00}, B^{06}, C^{12}, D^{18} >$ Remaining Time = $18 - 0 = 18$ Total Time = 18
- State 2: Prefix = $< A^{00} >$ ---- Postfix = $< B^{06}, C^{12}, D^{18} >$ Remaining Time = $18 - 0 = 18$ Total Time = 18
- State 3: Prefix = $< A^{00}, B^{06} >$ ---- Postfix = $< C^{12}, D^{18} >$ Remaining Time = $18 - 6 = 12$ Total Time = 18
- State 4: Prefix = $< A^{00}, B^{06}, C^{12} >$ ---- Postfix = $<, D^{18} >$ Remaining Time = $18 - 12 = 6$ Total Time = 18
- State 5: Prefix = $< A^{00}, B^{06}, C^{12}, D^{18} >$ ---- Postfix = $< >$ Remaining Time = $18 - 18 = 0$ Total Time = 18

This process is repeated for all other traces.

Predictive Process Mining

Different measurement functions can be used:

$$l_{elapsed}^{measure}(\sigma_1,\sigma_2) = \begin{cases} 0 & \text{if } \sigma_1 = \langle\rangle \\ \max_T(\sigma_1) - \min_T(\sigma_1) & \text{if } \sigma_1 \neq \langle\rangle \end{cases}$$

$$l_{total}^{measure}(\sigma_1,\sigma_2) = \begin{cases} 0 & \text{if } \sigma_1;\sigma_2 = \langle\rangle \\ \max_T(\sigma_1;\sigma_2) - \min_T(\sigma_1;\sigma_2) & \text{if } \sigma_1;\sigma_2 \neq \langle\rangle \end{cases}$$

$$l_{sojourn}^{measure}(\sigma_1,\sigma_2) = \begin{cases} 0 & \text{if } \sigma_1 = \langle\rangle \text{ or } \sigma_2 = \langle\rangle \\ \min_T(\sigma_2) - \max_T(\sigma_1) & \text{if } \sigma_1 \neq \langle\rangle \text{ and } \sigma_2 \neq \langle\rangle \end{cases}$$

Predictive Process Mining

- The functions listed above are all related to the duration of a process instance.
- It is also possible to provide completely other measurement functions

Prefix	Total Time	Remaining Time	Next Activity	Number of Activity	Will C take place?	When C Takes Place?
< >	18	18	A	4	1	12
< A^{00} >	18	18	B	4	1	12
< A^{00}, B^{06} >	18	12	C	4	1	6
< A^{00}, B^{06}, C^{12}>	18	6	D	4	0	-
< A^{00}, B^{06}, C^{12}, D^{18} >	18	0	-	4	0	-

Predictive Process Mining

- The functions listed above are all related to the duration of a process instance.
- It is also possible to provide completely other measurement functions

Prefix	Total Time	Remaining Time	Next Activity	Number of Activity	Will C take place?	When C Takes Place?
$\langle \rangle$	18	18	A	4	1	12
$\langle A^{00} \rangle$	18	18	B	4	1	12
$\langle A^{00}, B^{06} \rangle$	18	12	C	4	1	6
$\langle A^{00}, B^{06}, C^{12} \rangle$	18	6	D	4	0	-

Predictive Process Mining

- We can also generate functions that can be used as "Input Variables" of machine learning algorithms.

Prefix	Elapsed time	Number of Activity	N. Of Done Activities
< >	0	0	4
< A^{00} >	0	1	4
< A^{00}, B^{06} >	6	2	4
< A^{00}, B^{06}, C^{12}>	12	3	4

Predictive Process Mining

Repeat this process for all other traces:

Trace	Prefix	Total Time	Remaining Time	Next Activity	Number of Activity	Will C take place?	When C Takes Place?
1	$< A^{00} >$	18	18	B	3	1	12
2	$< A^{00} >$	16	16	B	3	0	–
3	$< A^{00} >$	24	24	–	1	0	–
4	$< A^{00} >$	19	19	C	2	0	–
...	...	...	...	...	...	...	...

Predictive Process Mining

Now we can consider "Prefix" and "Input Variable related functions" with all other case attributes (e.g. Age, Gender, …) and activity attributes (e.g. Resources, Cost) as ML Input Feature and other functions as ML Output Feature

Predictive Process Mining

Here are some different types of predictions that can be done using event logs:

- *Next Event Prediction*: Predicting the next event that is likely to occur.
- *Process Variant Prediction*: Predicting which process variant (e.g., different paths or branches) a case will follow.
- *Remaining Time Prediction*: Predicting the time remaining for the completion of an ongoing.
- *Case Completion Prediction*: Predicting whether a case will complete successfully or result in an exception/abortion.
- *Case Rework Prediction*: Predicting whether a case is likely to require rework (i.e., repeat certain steps or tasks).
- *Resource Prediction*: Predicting the resources (e.g., employees, machines) required to complete a case.
- *Anomaly Detection*: Identifying abnormal behavior or rare events in the process.
- *Case Outcome Prediction*: Predicting the outcome of a case (e.g., successful, delayed, rejected).
- *Customer Behavior Prediction*: Predicting customer behavior (e.g., churn prediction, purchase behavior).
- *Process Performance Prediction*: Predicting process performance metrics (e.g., cycle time, lead time) based on historical data.

Thanks