Project Title: Business Process Analysis using BPI Challenge 2019 Dataset

Student Name: Dauletbay Gulzat

Project Description:

The project aims to analyze and gain insights from the BPI Challenge 2019 dataset. The dataset represents a real-world business process in a specific application area. It includes information about various activities, timestamps, actors involved, and other relevant process data. The project will explore the dataset and extract valuable information to support decision-making and process improvement.

The data comes from a multinational company in the coatings and paints industry, based in the Netherlands. The company investigated the purchase order handling process across its 60 subsidiaries. The dataset includes information about purchase orders and their line items.

There are four types of flows related to the line items in the data:

1. 3-way matching, invoice after goods receipt: The value of the goods receipt message is matched against the value of an invoice receipt message. This flow requires both the GR-based flag and the Goods Receipt flags to be true.
2. 3-way matching, invoice before goods receipt: Goods receipt messages are required for these purchase items, but GR-based invoicing is not. Invoices can be entered before goods are received but are blocked until the goods arrive. Users or batch processes can unblock the invoices. Clearance is only allowed if the goods are received, and their value matches the invoice and the initial item value.
3. 2-way matching (no goods receipt needed): Separate goods receipt messages are not necessary for these items. The invoice value should match the initial value, either in full or partially until the purchase order value is consumed. Both the GR-based flag and the Goods Receipt flags are false for these items.
4. Invoice Receipt (IV) --> Match ValuesConsignment: This flow applies to items handled separately, without invoices on the purchase order level. The GR indicator is true, but the GR IV flag is false. Consignment items are not expected to have invoices based on their item type.

Multiple goods receipt messages and invoices can be associated with each purchased item. For example, paying rent may involve one purchase document item but multiple goods receipt messages and cleared invoices, each representing a fraction of the total amount. Similarly, logistical services may generate numerous goods receipt messages for one line item. Compliance requires matching the amounts of the line item, goods receipt messages, and invoices.

1. **Organizational Goals**:
    1.1. Summarize data
    1.2. Identify goals
2. **Knowledge Uplift Trail**
    2.1. Data preprocessing: Cleaning the dataset, handling missing values, and ensuring data quality.
    2.2. Exploratory Data Analysis (EDA): Analyzing the dataset's characteristics, identifying patterns, and visualizing process flow.
    2.3. Process Mining: Applying process mining techniques to discover the underlying process model, such as process maps and Petri nets.
    2.4. Conformance Analysis: Evaluating process performance metrics, such as cycle time, throughput, and bottlenecks.
3. **Project results**
4. **Conclusion**

1.1 Project

Summary of data:

- Total number of purchase documents: 76,349
- Total number of purchase items: 251,734
- Total number of cases (purchase document + purchase item combinations): 251,734
- Total number of events: 1,595,923
- Total number of activities: 42
- Total number of users: 627 (607 human users and 20 batch users)

Tools used in the project:

Colab for running Python code using libraries pandas and pm4py, also additional software like PMKT and Disco to observe the process map in various ways which makes it easy with different diagrams and filters.

Various **attributes** are recorded for each purchased item, but the **most important** ones are highlighted in the table below:

Table: Event Log

Attribute	**Description**
Case ID	Combination of the purchase document and purchase item IDs
Purchase Document	The anonymized ID of the purchase document
Purchase Item	The anonymized ID of the purchased item
Event ID	Unique identifier for each event within a case
Activity	Name of the activity performed in the event
Timestamp	A timestamp indicating when the event occurred
User	Identifier of the user who performed the activity (human user or batch user)

1.2 Goals

1.2.1. Goals for the project:

- Analyze the BPI Challenge 2019 dataset from a multinational company in the coatings and paints industry.
- Gain insights into the purchase order handling process across the company's subsidiaries.
- Extract valuable information to support decision-making and process improvement.

1.2.2. Research Questions to Address BPI Challenge:

- Are there any loops or repetitions in the process?
- Are there any bottlenecks or inefficiencies in the process?
- Can we predict the completion time of a process instance based on its attributes?
- How does the process flow differ for each of the four types of flows related to the line items?
- Are there any patterns or variations in the process that can be observed from the data?

2.1 Data preprocessing

Data preprocessing: Cleaning the dataset, handling missing values, and ensuring data quality.

The initial phase of data preprocessing involves dropping duplicates from the dataset. Before checking for duplicates, there were 11,973 unique variants and 251,734 cases. After removing duplicates, the number of unique variants decreased to 11,319 while maintaining the same number of cases.

The second step includes filtering out incomplete cases that do not reach the "create purchase order item" stage. Following this filtering process, the number of unique variants was further reduced to 11,311, representing a decrease of 5.5%. The number of cases decreased to 247,287, which accounts for a reduction of 1.7%.

Next, the time duration considered for analysis was limited to the last three years. After applying this time constraint, the number of unique variants decreased to 11,267, which corresponds to a reduction of 5.7%. The number of cases decreased to 247209, representing a decrease of 1.8%.

2.2 Exploratory Data Analysis

The basic steps help to clean data and reduce it to N%. However, working with 10 thousand variants challengeable task, and also it is hard to identify bottlenecks and analyze throughputs. The solution is to divide the dataset into four types of flows related to the line items in the data.

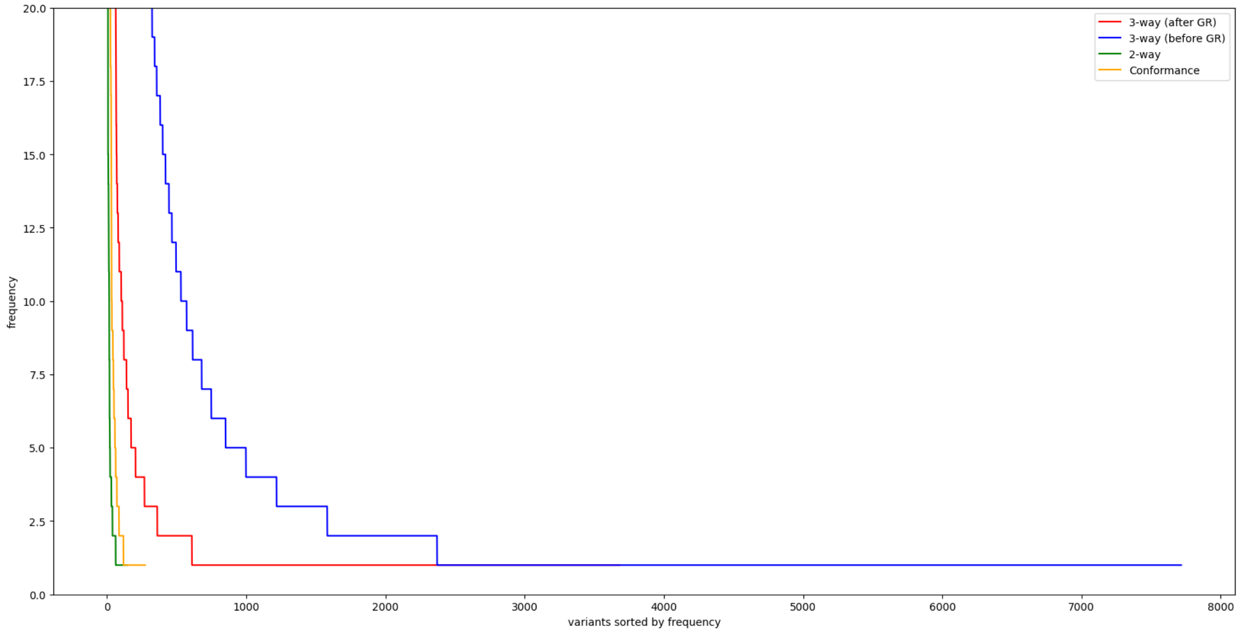
Illustrations of key differences of types in tables

The table provides a more detailed representation of the categories, their attributes, and the matching process for each category, with separate columns indicating whether the goods receipt value, invoice value, and creation value are matched.

Category Name	**GR-based Flag**	**Goods Receipt Flags**	**Match Goods Receipt Value**	**Match Invoice Value**	**Match Creation Value**	**3-way matching, invoice after**
3-way matching	True	True	Yes	Yes	Yes	Yes
3-way matching, invoice before	False	True	Yes	User or Batch Process	Yes	Yes
2-way matching	False	False	N/A	Yes	Yes	N/A
Consignment	True	False	N/A	N/A	N/A	N/A

Variant analysis for each type:

- 3-ways matching after GR variants: 3682
- 3-ways matching before GR variants: 7717
- 2-ways matching variants: 147
- Conformance variants: 274

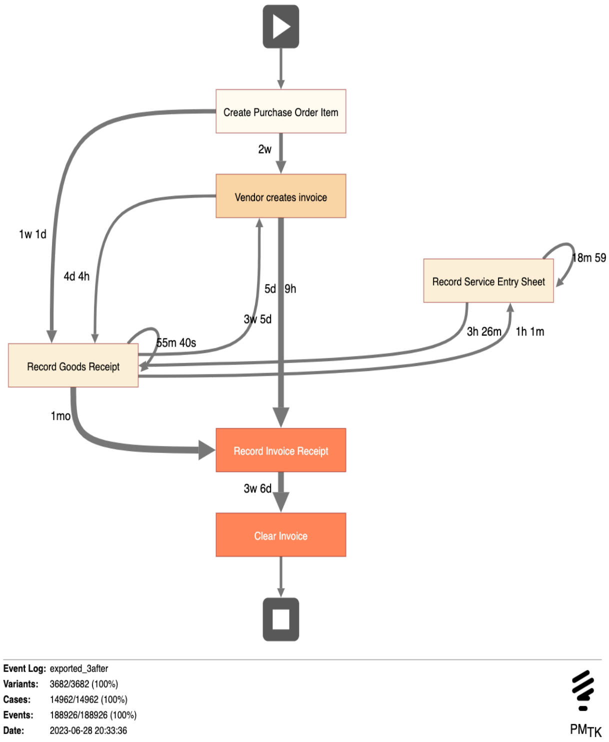


Process Maps

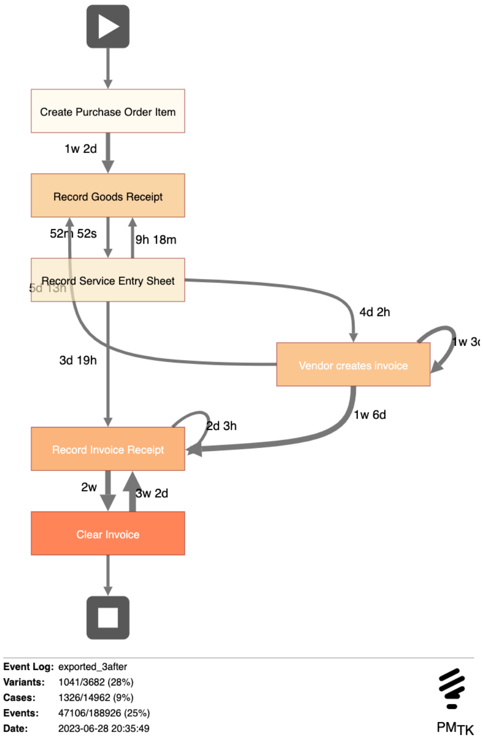
Process maps visually represent the sequence and dependencies of activities in a flow, allowing you to identify patterns, bottlenecks, and variations.

- 3-ways matching after GR variants:

Two pictures from pmkt show us the difference in all data and filtering by reworking have a significant difference between a number of cases and variants and events.



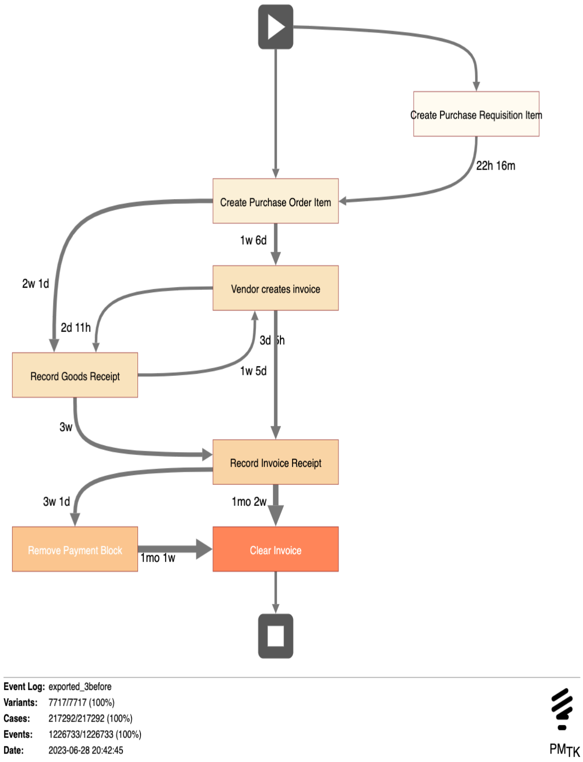
Event Log: exported_3after
Variants: 3682/3682 (100%)
Cases: 14962/14962 (100%)
Events: 188926/188926 (100%)
Date: 2023-06-28 20:33:36



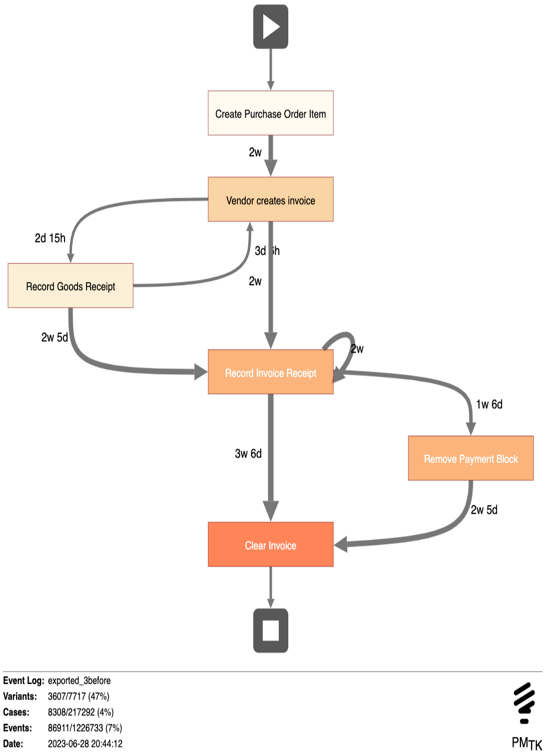
Event Log: exported_3after
Variants: 1041/3682 (28%)
Cases: 1326/14962 (9%)
Events: 47106/188926 (25%)
Date: 2023-06-28 20:35:49

- 3-ways matching before GR variants: 7717

Overall the situation with reworking is repeated again, with different percentages reducing in cases, and variants.

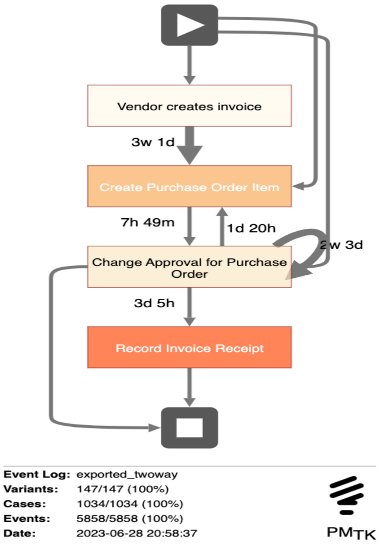


Event Log: exported_3before
Variants: 7717/7717 (100%)
Cases: 217292/217292 (100%)
Events: 1226733/1226733 (100%)
Date: 2023-06-28 20:42:45



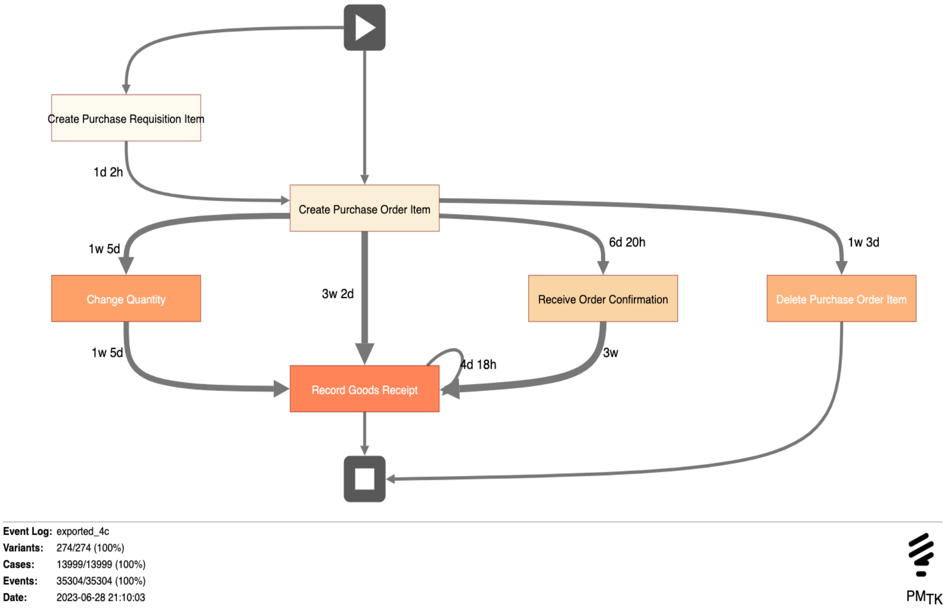
Event Log: exported_3before
Variants: 3607/7717 (47%)
Cases: 8308/217292 (4%)
Events: 86911/1226733 (7%)
Date: 2023-06-28 20:44:12

- 2-ways matching



Event Log: exported_twoway
Variants: 147/147 (100%)
Cases: 1034/1034 (100%)
Events: 5858/5858 (100%)
Date: 2023-06-28 20:58:37

- Consignment


Create Purchase Requisition Item
1d 2h
Create Purchase Order Item
1w 5d
6d 20h
1w 3d
Change Quantity
3w 2d
Receive Order Confirmation
Delete Purchase Order Item
1w 5d
4d 18h
3w
Record Goods Receipt
PMTK

Event Log: exported_4c
Variants: 274/274 (100%)
Cases: 13999/13999 (100%)
Events: 35304/35304 (100%)
Date: 2023-06-28 21:10:03

There are no huge optimizations by time filtering reworks or too distorted the process maps

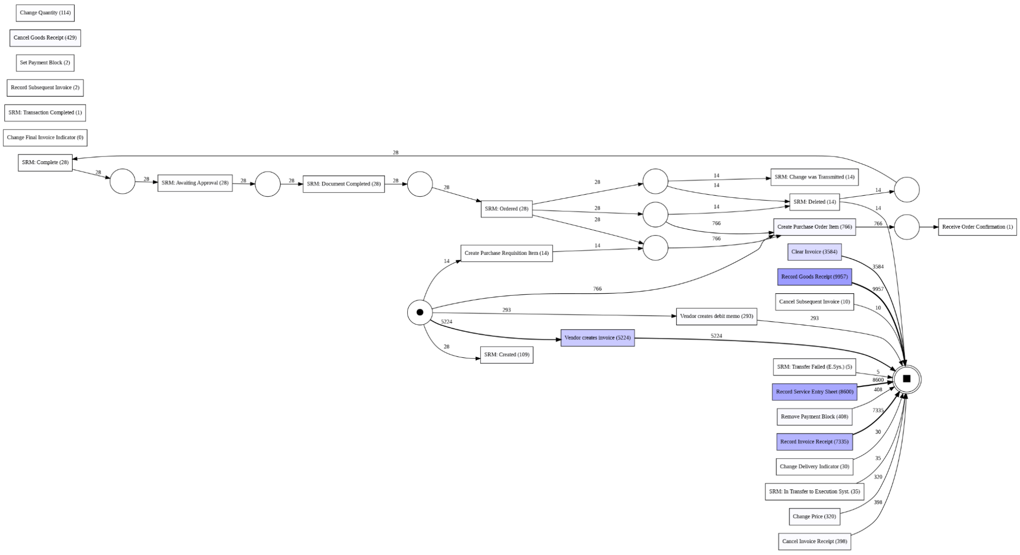
2.3 Process Discovery

3-way matching after GR

To visualization reduce the reworks on step "Record Invoice Receipt"

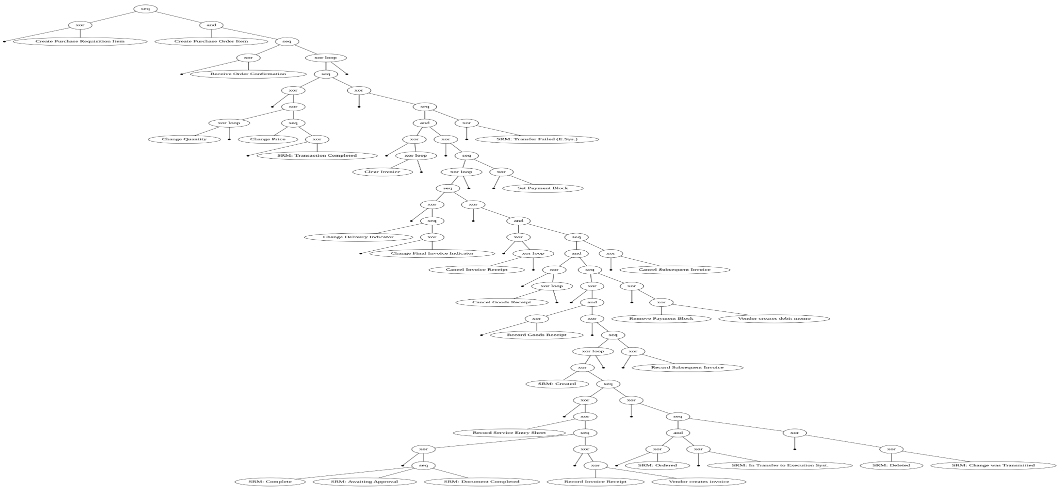
Below are all diagrams Alpha mining, Inductive Mining with Petri nets also heuristic mining with a flow diagram for the first type. The detailed image will be provided on GitHub repository.

1. Alpha mining- discovers causal relationships between events



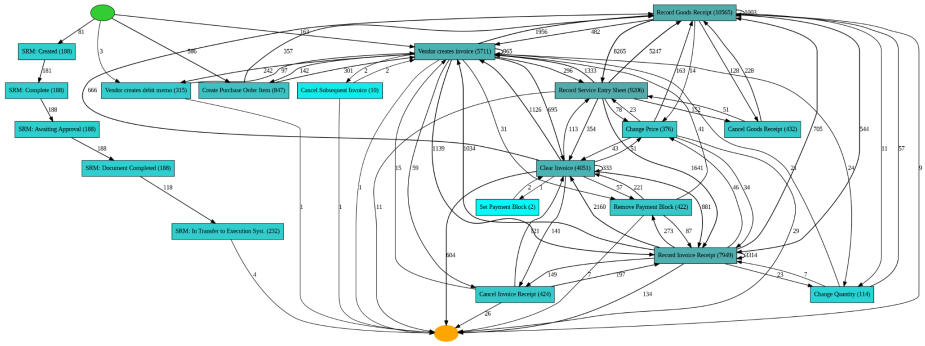
Change Quantity (114)
Cancel Goods Receipt (429)
Set Payment Block (2)
Record Subsequent Invoice (2)
SRM: Transaction Completed (1)
Change Final Invoice Indicator (0)
SRM: Complete (28)
SRM: Awaiting Approval (28)
SRM: Document Completed (28)
SRM: Ordered (28)
SRM: Change was Transmitted (14)
SRM: Deleted (14)
Create Purchase Order Item (766)
Receive Order Confirmation (1)
Create Purchase Requisition Item (14)
Clear Invoice (3584)
Record Goods Receipt (9957)
Cancel Subsequent Invoice (10)
Vendor creates debit memo (293)
Vendor creates invoice (5224)
SRM: Created (109)
SRM: Transfer Failed (E.Sys.) (5)
Record Service Entry Sheet (8600)
Remove Payment Block (408)
Record Invoice Receipt (7335)
Change Delivery Indicator (30)
SRM: In Transfer to Execution Syst. (35)
Change Price (320)
Cancel Invoice Receipt (398)
28
14
766
293
5224
3584
9957
10
5
8600
408
7335
30
35
320
398

2. Inductive mining representation in the tree

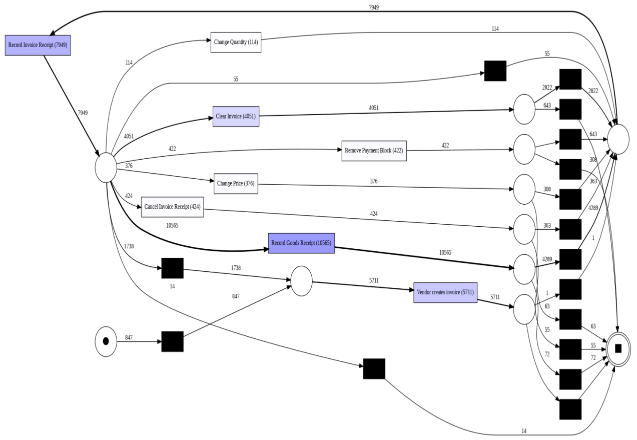


seq
xor
and
xor loop
Create Purchase Requisition Item
Create Purchase Order Item
Receive Order Confirmation
Change Quantity
Change Price
SRM: Transaction Completed
SRM: Transfer Failed (E.Sys.)
Clear Invoice
Set Payment Block
Change Delivery Indicator
Change Final Invoice Indicator
Cancel Invoice Receipt
Cancel Subsequent Invoice
Cancel Goods Receipt
Remove Payment Block
Vendor creates debit memo
Record Goods Receipt
Record Subsequent Invoice
SRM: Created
Record Service Entry Sheet
SRM: Ordered
SRM: In Transfer to Execution Syst.
SRM: Deleted
SRM: Change was Transmitted
SRM: Complete
SRM: Awaiting Approval
SRM: Document Completed
Record Invoice Receipt
Vendor creates invoice

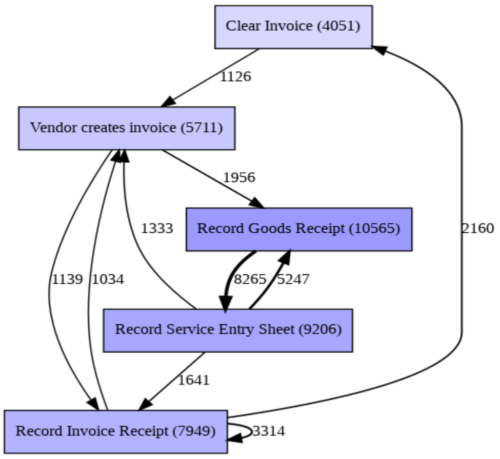
3. Heuristic mining relations representation


Record Goods Receipt (10565)
SRM: Created (188)
Vendor creates invoice (5711)
SRM: Complete (188)
Vendor creates debit memo (315)
Create Purchase Order Item (847)
Cancel Subsequent Invoice (10)
Record Service Entry Sheet (9206)
SRM: Awaiting Approval (188)
Change Price (376)
Cancel Goods Receipt (432)
SRM: Document Completed (188)
Clear Invoice (4051)
Set Payment Block (2)
Remove Payment Block (422)
SRM: In Transfer to Execution Syst. (232)
Record Invoice Receipt (7949)
Cancel Invoice Receipt (424)
Change Quantity (114)

4. Heuristic with frequency representation

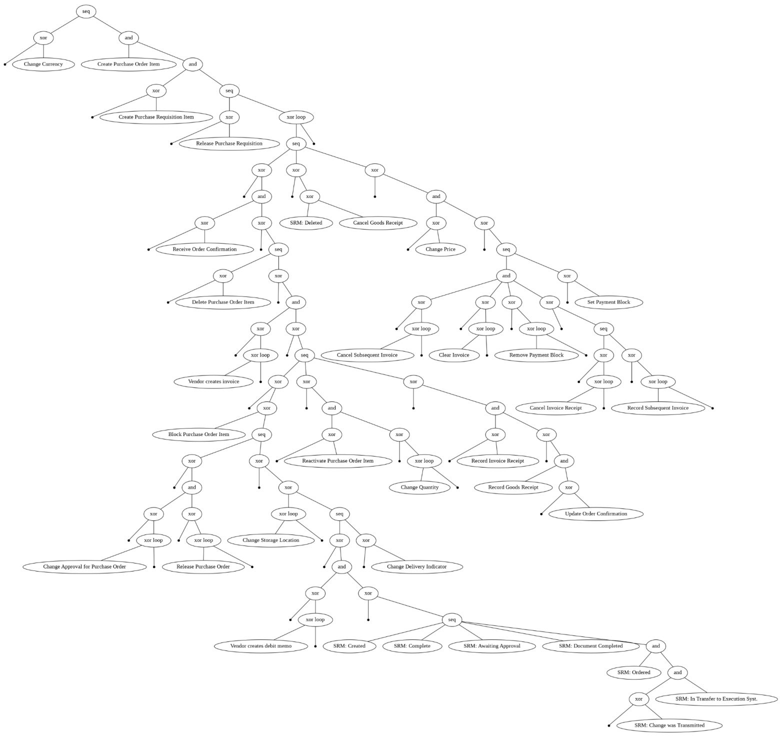

7949
Record Invoice Receipt (7949)
Change Quantity (114)
114
Clear Invoice (4051)
4051
Remove Payment Block (422)
422
Change Price (376)
376
Cancel Invoice Receipt (424)
424
Record Goods Receipt (10565)
10565
Vendor creates invoice (5711)
5711
1738
847
14

5. Graph from logs

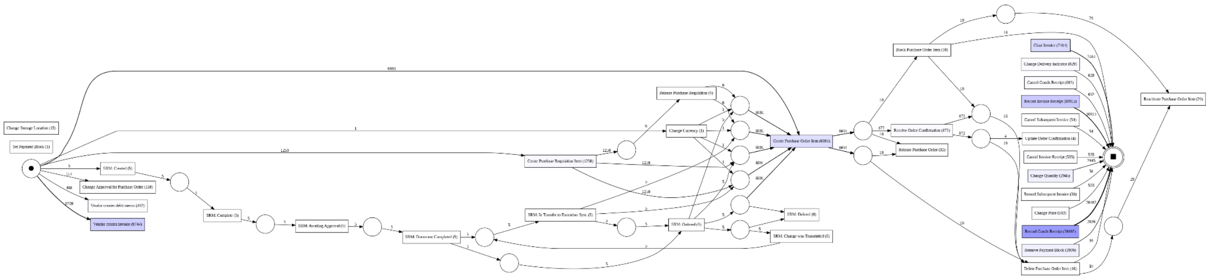

Clear Invoice (4051)
1126
Vendor creates invoice (5711)
1956
1333
Record Goods Receipt (10565)
2160
1139
1034
8265
5247
Record Service Entry Sheet (9206)
1641
Record Invoice Receipt (7949)
3314

3-way matching before GR

1. **Inductive mining representation in the tree**

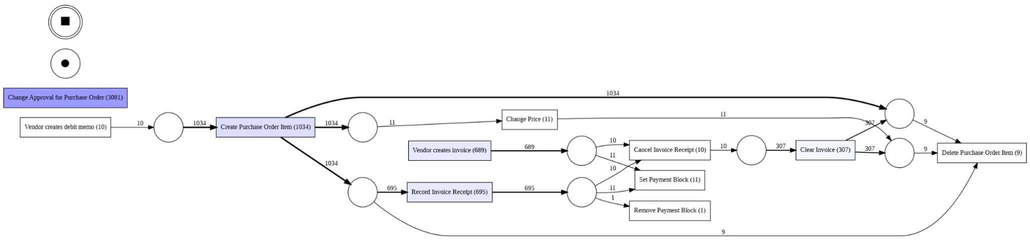

seq
xor
and
Change Currency
Create Purchase Order Item
Create Purchase Requisition Item
Release Purchase Requisition
xor loop
Receive Order Confirmation
SRM: Deleted
Cancel Goods Receipt
Change Price
Delete Purchase Order Item
Set Payment Block
Cancel Subsequent Invoice
Clear Invoice
Remove Payment Block
Vendor creates invoice
Cancel Invoice Receipt
Record Subsequent Invoice
Block Purchase Order Item
Reactivate Purchase Order Item
Record Invoice Receipt
Change Quantity
Record Goods Receipt
Update Order Confirmation
Change Approval for Purchase Order
Release Purchase Order
Change Storage Location
Change Delivery Indicator
Vendor creates debit memo
SRM: Created
SRM: Complete
SRM: Awaiting Approval
SRM: Document Completed
SRM: Ordered
SRM: In Transfer to Execution Syst.
SRM: Change was Transmitted

2. **The alpha mining graph represents relations between processes and is highlighted by frequency**

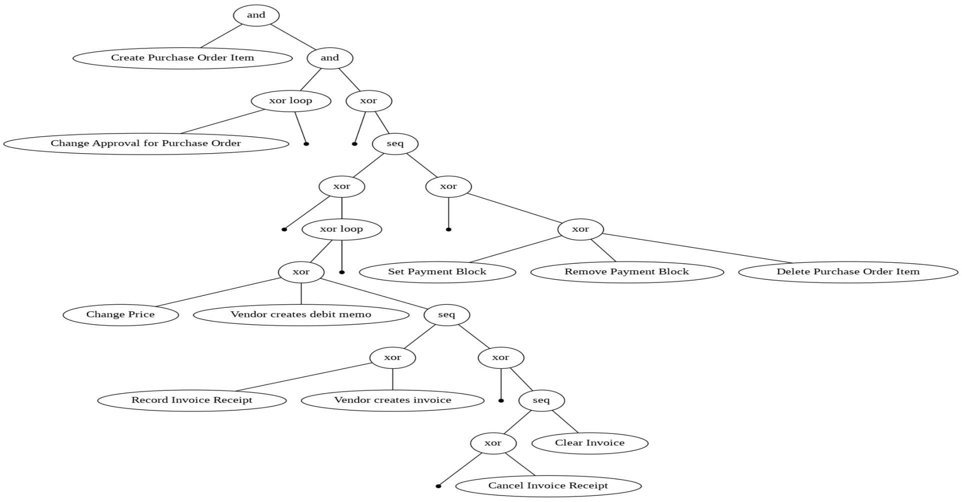

Change Storage Location (15)
Set Payment Block (1)
SRM: Created (5)
Change Approval for Purchase Order (118)
Vendor creates debit memo (492)
Vendor creates invoice (9744)
SRM: Complete (5)
SRM: Awaiting Approval (5)
SRM: Document Completed (5)
Create Purchase Requisition Item (1250)
SRM: In Transfer to Execution Syst. (5)
Release Purchase Requisition (6)
Change Currency (1)
SRM: Ordered (5)
SRM: Deleted (0)
SRM: Change was Transmitted (5)
Create Purchase Order Item (6691)
Block Purchase Order Item (18)
Receive Order Confirmation (872)
Release Purchase Order (32)
Clear Invoice (7484)
Change Delivery Indicator (829)
Cancel Goods Receipt (697)
Record Invoice Receipt (10815)
Cancel Subsequent Invoice (54)
Update Order Confirmation (4)
Cancel Invoice Receipt (535)
Change Quantity (2945)
Record Subsequent Invoice (38)
Change Price (503)
Record Goods Receipt (20492)
Remove Payment Block (2036)
Delete Purchase Order Item (16)
Reactivate Purchase Order Item (29)
6691
1
1250
5
114
488
9709
18
872
4
16
29

2-way matching

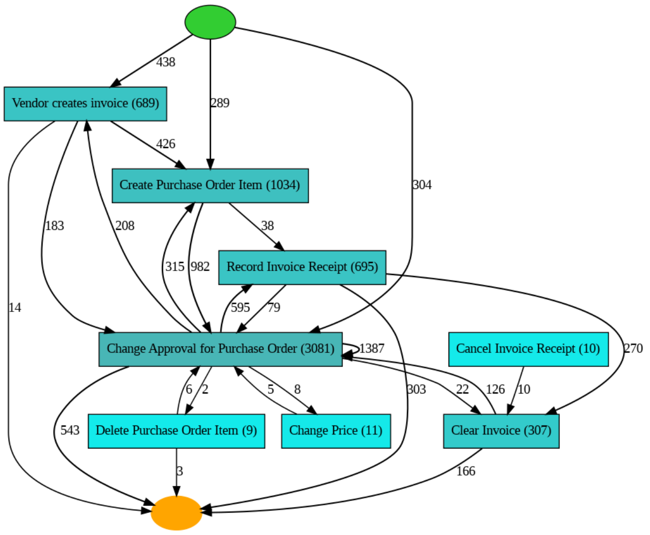
1. Alpha mining- discovers causal relationships between events


Change Approval for Purchase Order (3081)
Vendor creates debit memo (10)
10
1034
Create Purchase Order Item (1034)
1034
1034
11
1034
Change Price (11)
11
Vendor creates invoice (689)
689
10
11
Cancel Invoice Receipt (10)
10
307
Clear Invoice (307)
307
307
9
9
Delete Purchase Order Item (9)
10
Set Payment Block (11)
695
Record Invoice Receipt (695)
695
11
1
Remove Payment Block (1)
9

2. Inductive mining visualization

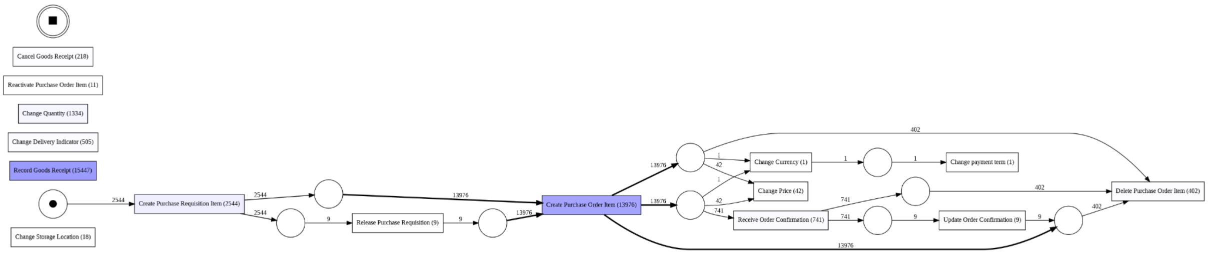


3. Heuristic mining visualization


438
289
304
Vendor creates invoice (689)
426
Create Purchase Order Item (1034)
183
208
38
315
982
Record Invoice Receipt (695)
14
595
79
Change Approval for Purchase Order (3081)
1387
Cancel Invoice Receipt (10)
270
6
2
5
8
303
22
126
10
543
Delete Purchase Order Item (9)
Change Price (11)
Clear Invoice (307)
3
166

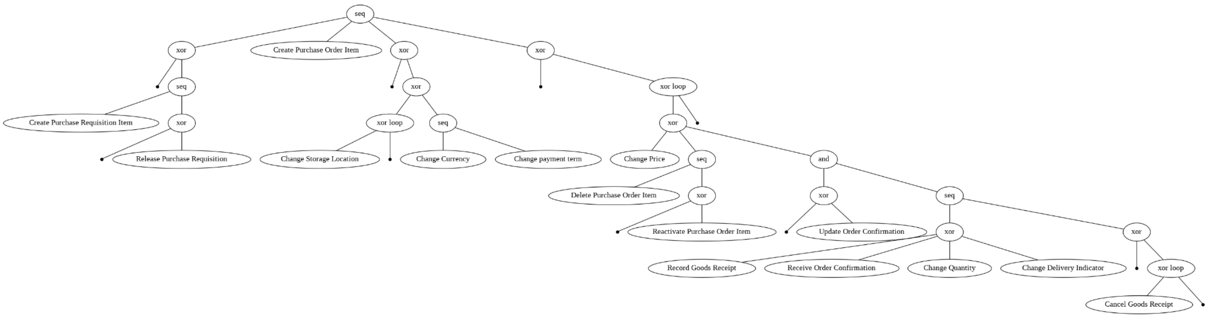
Consginments

1. Alpha mining- discovers causal relationships between events



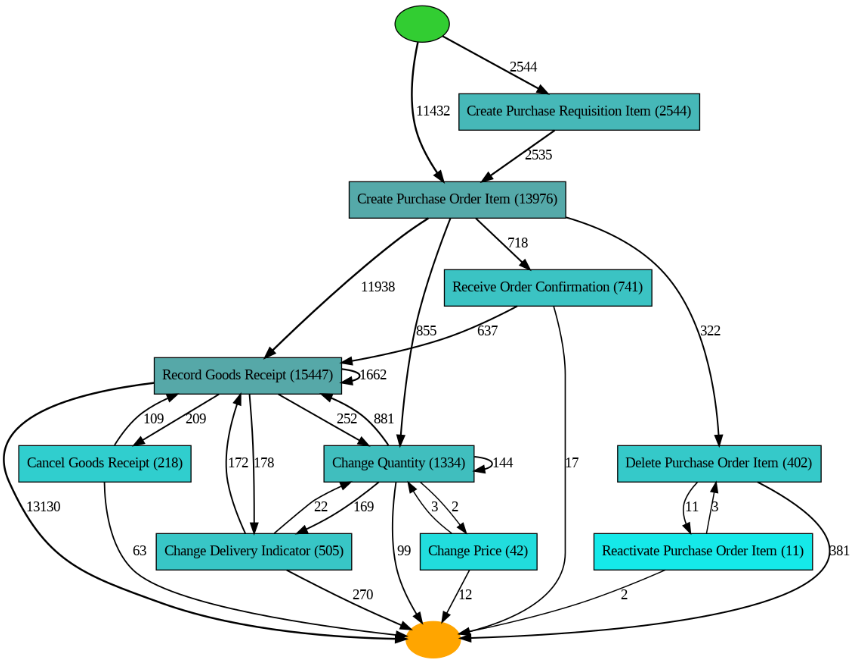
Cancel Goods Receipt (218)
Reactivate Purchase Order Item (11)
Change Quantity (1334)
Change Delivery Indicator (505)
Record Goods Receipt (15447)
Change Storage Location (18)
2544
Create Purchase Requisition Item (2544)
2544
2544
9
Release Purchase Requisition (9)
9
13976
13976
Create Purchase Order Item (13976)
13976
13976
1
42
1
42
741
Change Currency (1)
Change Price (42)
1
1
Change payment term (1)
402
741
741
Receive Order Confirmation (741)
9
Update Order Confirmation (9)
9
402
402
Delete Purchase Order Item (402)
13976

2. Inductive mining visualization



seq
xor
Create Purchase Order Item
xor
xor
seq
xor
xor loop
Create Purchase Requisition Item
xor
xor loop
seq
xor
Release Purchase Requisition
Change Storage Location
Change Currency
Change payment term
Change Price
seq
and
Delete Purchase Order Item
xor
xor
seq
Reactivate Purchase Order Item
Update Order Confirmation
xor
xor
Record Goods Receipt
Receive Order Confirmation
Change Quantity
Change Delivery Indicator
xor loop
Cancel Goods Receipt

3. Heuristic mining visualization



2544
11432
Create Purchase Requisition Item (2544)
2535
Create Purchase Order Item (13976)
718
Receive Order Confirmation (741)
11938
855
637
322
Record Goods Receipt (15447)
1662
109
209
252
881
Cancel Goods Receipt (218)
172
178
Change Quantity (1334)
144
17
Delete Purchase Order Item (402)
13130
22
169
3
2
11
3
63
Change Delivery Indicator (505)
99
Change Price (42)
Reactivate Purchase Order Item (11)
381
270
12
2

2.4 Conformance Analysis

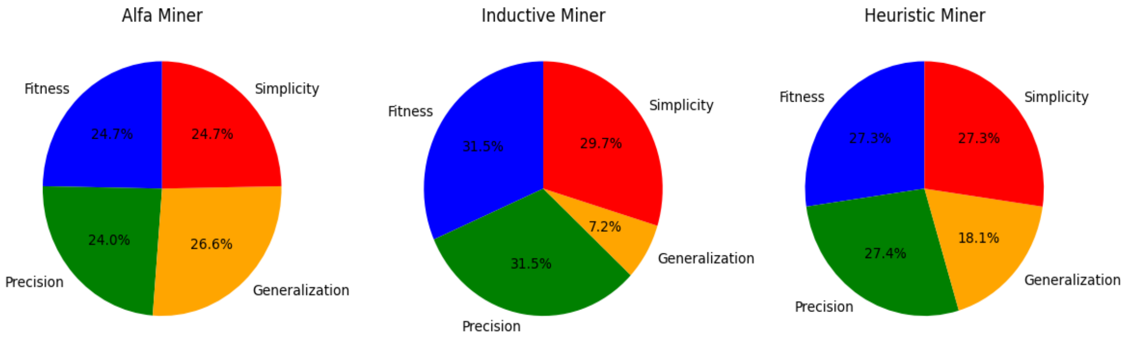
3-way matching after GR

Metric	Fitness	Precision	Generalization	Simplicity
Alpha Mining	'perc_fit_traces': 0.0, 'average_trace_fitness': 0.219, 'log_fitness':0.163, 'percentage_of_fitting_traces': 0.0	0.152	0.776	1.0
Inductive Mining	'perc_fit_traces': 4.959, 'average_trace_fitness': 0.978, 'log_fitness':0.971, 'percentage_of_fitting_traces': 4.959	0.168	0.890	0.610
Heuristic Mining	'perc_fit_traces': 0.0, 'average_trace_fitness': 0.647, 'log_fitness':0.671, 'percentage_of_fitting_traces': 0.0	0.540	0.779	0.574

In summary, both Inductive Mining and Heuristic Mining outperform Alpha Mining in terms of fitness and precision, indicating better conformance to the log data and more accurate capturing of behavior. However, Inductive Mining shows a higher level of simplicity compared to Heuristic Mining, while Heuristic Mining demonstrates a higher level of generalization.

3-way matching before GR

Metric	Fitness	Precision	Generalization	Simplicity
Alfa Miner	'perc_fit_traces': 0.0, 'average_trace_fitness': 0.265, 'log_fitness':0.258, 'percentage_of_fitting_traces': 0.0	0.265	0.258	0.285
Inductive Miner	'perc_fit_traces':100.0, 'average_trace_fitness': 1.0000, 'log_fitness':1.0000, 'percentage_of_fitting_traces': 100.0	1.000	1.000	0.228
Heuristic Miner	'perc_fit_traces': 10.172 average_trace_fitness':0.942, 'log_fitness':0.946, 'percentage_of_fitting_traces': 10.172	0.942	0.946	0.625



In a comparative analysis of the metrics and visualized pie charts, Inductive Miner stands out with a perfect fitness score of 1.000 and a high precision score of 0.228, indicating strong conformance to the log data. Alfa Miner shows moderate fitness and precision scores, with a fitness score of 0.265 and a precision score of 0.285. Heuristic Miner demonstrates good fitness and precision scores of 0.942 and 0.625, respectively, but falls slightly behind Inductive Miner in terms of overall conformance.

2-way matching

title	Fitness	Precision	Generalization	Simplicity
Alpha Miner	'perc_fit_traces': 0.0, 'average_trace_fitness': 0.1585, 'log_fitness':0.1856, 'percentage_of_fitting_traces': 0.0	0.887	0.750	0.9130
Inductive Miner	'perc_fit_traces':100.0, 'average_trace_fitness': 1.0000, 'log_fitness':1.0000, 'percentage_of_fitting_traces': 100.0	0.344	0.837	0.5738
Heuristic Miner	'perc_fit_traces': 0.0, ' average_trace_fitness': 0.4769, 'log_fitness':0.3540, 'percentage_of_fitting_traces': 0.0	0.335	0.750	1.0000

In summary for 2-ways matching, the Inductive Miner algorithm achieves a perfect fitness score, indicating a higher level of conformance than the Alpha Miner and Heuristic Miner. However, the Alpha Miner algorithm performs well in terms of precision and simplicity. The Heuristic Miner algorithm has a moderate fitness score and the highest simplicity value.

Consignment

Algorithm	perc_fit_traces	average_trace_fitness	log_fitness	percentage_of_fitting_traces	precision	generalization	simplicity
Alpha	0.0	0.0972	0.127	0.0	0.1429	0.758	1
Inductive	100.0	1.0	1.0	100.0	0.8609	0.798	0.628
Heuristic	81.855	0.9610	0.955	81.855	0.9993	0.954	0.647

In the table, the metrics for each algorithm are presented:

- perc_fit_traces: Percentage of fitting traces.
- average_trace_fitness: Average trace fitness.
- log_fitness: Log fitness.
- percentage_of_fitting_traces: Percentage of fitting traces.
- precision: Precision metric.
- generalization: Generalization metric.
- simplicity: Simplicity metric.

Based on these metrics, the Inductive Miner generally outperforms the Alpha Miner and Heuristic Miner in terms of conformance and precision for type consignment. However, the Heuristic Miner shows competitive performance with high precision and moderate conformance.

3. Project Results

Each step of the project was implemented using Python and the PM4Py, pandas libraries. The code cells in the submitted Colab project are referenced for each analytical step. The detailed results are included:

1. Preprocessed dataset: Cleaned and prepared the dataset for analysis.
2. Process maps: Visual representation of the discovered process model.
3. Process discovering: Analysing with 3 types of mining
4. Conformance analyzing: based on the previous step compare the best mining for each

4. Conclusion

In conclusion, the project "Business Process Analysis using BPI Challenge 2019 Dataset" aimed to analyze and gain insights from the BPI Challenge 2019 dataset, which represents a real-world purchase order handling process in the coatings and paints industry. The project successfully achieved its goals of analyzing the dataset, identifying patterns, and extracting valuable information to support decision-making and process improvement.

The project began with data preprocessing, which involved cleaning the dataset, handling missing values, and ensuring data quality. Duplicates and incomplete cases were removed, and a time constraint was applied to focus on the last three years of data.

Exploratory Data Analysis (EDA) was conducted to understand the dataset's characteristics and visualize process flows. The dataset was divided into four types of flows related to the line items, namely 3-way matching after goods receipt, 3-way matching before goods receipt, 2-way matching, and consignment. Process maps were created for each type, providing a visual representation of the sequence and dependencies of activities.

Process mining techniques were then applied to discover the underlying process models. Three mining algorithms, namely Alpha Miner, Inductive Miner, and Heuristic Miner, were used for process discovery. The results showed that Inductive Miner generally outperformed the other algorithms in terms of fitness and precision, indicating better conformance to the log data and more accurate capturing of behavior. However, Heuristic Miner demonstrated a higher level of simplicity and competitive performance with high precision.

Conformance analysis was conducted to evaluate the discovered process models' conformance to the log data. Inductive Miner showed a strong conformance to the log data, while Alpha Miner and Heuristic Miner demonstrated moderate conformance.

The project's results provided valuable insights into the purchase order handling process, including identifying bottlenecks, inefficiencies, and variations in the process. The process maps and conformance analysis highlighted areas where the process could be improved or optimized.

Overall, the project successfully analyzed the BPI Challenge 2019 dataset, provided insights into the purchase order handling process, and laid the foundation for process improvement initiatives. The findings can support decision-making, identify areas for optimization, and contribute to enhancing the efficiency and effectiveness of the process in the coatings and paints industry.

References

1. Materials and pieces of code from course BIS
2. https://medium.com/@c3_62722/process-mining-with-python-tutorial-a-healthcare-application-part-2-4cf57053421f
3. Github link for the project: https://github.com/gulzat-dev/BIS
4. PMKT, COLAB, DISCO, PM4PY, etc.