

**UNIVERSITY OF TIRANA**

Faculty of Natural Sciences

Department of Computer Science

**Course :** Process Mining

**Project Title:** Optimizing the Application Process for Administrative Positions of the DAP Portal

**Instructor :** Msc.Anxhela Kosta

**Student :** Andeta Zeqiri

**1. Introduction**

This project describes the modeling and analysis of the application process for administrative positions using the DAP portal. The study encompasses multiple analytical approaches including Business Process Model and Notation (BPMN), decision trees, Petri nets, clustering algorithms, and process discovery algorithms to understand, model, and optimize the recruitment workflow. The most important goal is to discover bottlenecks, streamline applicant flow, and increase transparency as well as optimize efficiency in processes.

**Project Objectives**

The primary objectives of this project are to:

* Model and analyze the complete recruitment process of the DAP portal
* Apply process mining algorithms to discover patterns and bottlenecks
* Utilize clustering techniques for behavioral analysis and outcome classification
* Compare different process discovery algorithms for optimal model fitness
* Identify areas for process improvement and optimization
* Enhance transparency and efficiency in the application process

**Problem Statement**

The DAP portal serves as the official platform for recruitment in public administration. While it digitalizes the application process, challenges remain in efficient candidate processing, decision transparency, and timely notifications. The multi-phase selection process can create confusion, delays, and lack of clarity for applicants. This project provides a comprehensive analysis to uncover data-driven improvement opportunities.

**Methodology and Technical Implementation**

Tools and Technologies Used

* **PM4Py**: Process mining library for Python
* **Pandas**: Data manipulation and analysis
* **Scikit-learn**: Machine learning algorithms (K-Means clustering)
* **Matplotlib/Seaborn**: Data visualization
* **Power Automate Process Mining**: Petri Net modeling
* **BPMN Modeling Tools**: Process diagram creation

Data Quality and Validation

* Comprehensive data cleaning and preprocessing
* Timestamp validation and conversion
* Event log structure verification
* Cross-validation with multiple algorithms
* Manual verification of process logic

**2. Process Modeling and Representation**

**BPMN Process Model**

The Business Process Model and Notation (BPMN) diagram illustrates the complete application journey from portal access to final candidate selection. The process includes:

**Key Process Elements:**

* **Start Events**: Portal access, position selection
* **Decision Gateways**: Application deadline checks, document verification, criteria fulfillment
* **Intermediate Tasks**: Document review, eligibility assessment, appeals handling, examination phases
* **End Events**: Successful hiring or rejection notifications

**Process Flow Overview:**

1. **Portal Access and Position Selection**: Applicants access the DAP portal and select available positions
2. **Document Preparation and Submission**: Required documents are prepared and submitted before deadline
3. **Document Review**: System checks for missing documents and notifies applicants
4. **Eligibility Assessment**: Verification of criteria fulfillment with appeal process available
5. **Phase 1 - Written Examination**: Qualifying exam with minimum score requirements
6. **Phase 2 - Oral Interview**: Final selection phase for successful candidates
7. A diagram of a flowchart

   AI-generated content may be incorrect.**Winner Selection and Communication**: Institution contacts selected candidates for position acceptance

**Decision Tree Analysis**

The decision tree provides a structured approach to the application process, breaking down complex decisions into manageable steps:

**Key Decision Points:** A diagram of a company

AI-generated content may be incorrect.

* Application deadline status
* Document completeness verification
* Criteria fulfillment assessment
* Examination performance evaluation
* Final selection and acceptance decisions

The decision tree reveals critical branching points where applications can be rejected or proceed to subsequent phases, highlighting the importance of each evaluation stage.

**Petri Net Representation**

The Petri Net formal model was developed following these specifications:

* **Places**: Represent states before or after activities
* **Transitions**: Actual process activities
* **Arcs**: Show process flow connections

The Petri Net was designed using both manual creation and Power Automate Process Mining tools, ensuring consistency with the BPMN model while providing a formal mathematical representation suitable for analysis.

A diagram of a flowchart

AI-generated content may be incorrect.A diagram of a software process

AI-generated content may be incorrect.

**3. Data Generation and Event Logs**

**Simulated Event Log Creation**

A comprehensive CSV dataset was manually created to simulate realistic event logs based on the BPMN and Petri Net models. The dataset structure includes:

**Data Schema:**

* **CaseID**: Unique identifier for each application process
* **Activity**: Specific task being performed
* **InitialStatus**: State before the activity
* **FinalStatus**: State after the activity
* **ProcessFlow**: Description of the activity's role in the process
* **Timestamp**: When the activity occurred

**Dataset Characteristics:**

* 8 cases (applications) representing different process outcomes
* Various activities spanning the complete application lifecycle
* Realistic timestamps reflecting actual process durations
* Multiple process variants including successful and unsuccessful applications

**Event Log Processing**

The event logs were processed using PM4Py library with the following transformations:

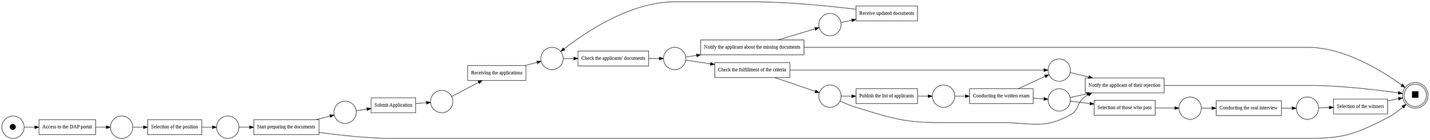
* Timestamp conversion to proper datetime format
* Column renaming for PM4Py compatibility
* Dataframe to event log conversion
* Data validation and cleaning procedures

**4. Process Discovery and Analysis**

**Alpha Miner Algorithm Application**

After applying the Alpha Miner algorithm to the event logs, I was able to successfully discover the process model. The main activities identified in the process were:

* Submit Application
* Review Application
* Accept Application
* Reject Application

What stood out to me was the clear branching between accepted and rejected applications, which shows how the process can take different paths. The overall flow of the process seemed consistent, with a specific order of steps. Also, the algorithm did a good job of capturing variations in how the process is actually carried out in different cases.

**Algorithm Comparison Analysis**

In this section, a detailed comparison was made between three commonly used process discovery algorithms: Inductive Miner, Alpha Miner, and Heuristic Miner. Each algorithm was evaluated based on its fitness, precision, and overall performance.

| **Algorithm** | **Fitness** | **Precision** | **Performance** |
| --- | --- | --- | --- |
| Inductive Miner | 1.0000 | 0.7845 | Good |
| Alpha Miner | 0.9006 | 0.9759 | Very Good |
| Heuristic Miner | 1.0000 | 0.9785 | Best Overall |

The **Heuristic Miner** algorithm delivered the strongest performance overall. It achieved a perfect fitness score of **1.0000**, indicating that it was able to reproduce all the behavior observed in the event logs. It also had the **highest precision** score of **0.9785**, meaning it produced fewer behaviors that were not seen in the logs.

The **Alpha Miner** showed a well-balanced performance, with a good fitness score of **0.9006** and a high precision score of **0.9759**. This suggests that while it slightly missed some of the behavior from the logs, it still maintained a high level of accuracy in terms of model correctness.

The **Inductive Miner** achieved a perfect fitness score of **1.0000**, similar to the Heuristic Miner, but its **precision was lower**, at **0.7845**. This means it allowed for more behavior that was not present in the logs, which can reduce the reliability of the discovered model.

In summary, while each algorithm has its strengths, the **Heuristic Miner** stands out as the best-performing algorithm due to its ability to combine both perfect fitness and the highest precision.

A screenshot of a computer screen

AI-generated content may be incorrect.

**5. K-Means Clustering Analysis**

Case-level features were extracted for clustering analysis:

**Extracted Features:**

1. **Duration** (days): Total time from first to last activity
2. **Number of Events**: Total activity count per case
3. **Number of Unique Activities**: Count of distinct activities
4. **Average Time Between Activities**: Mean interval between consecutive activities
5. **Activity Transitions**: Number of process flow changes

**Optimal Cluster Determination**

**A screenshot of a chart

AI-generated content may be incorrect.A graph with a line

AI-generated content may be incorrect.A graph of a function

AI-generated content may be incorrect.**Using the Elbow Method, **6 clusters** were determined as optimal, based on the sum of squared distances analysis.

**A graph with colored dots

AI-generated content may be incorrect.**

**Cluster Analysis Results**

| **Cluster** | **Count** | **Duration (days)** | **Num Events** | **Unique Activities** | **Avg Time Between (days)** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 0.11 | 7.5 | 7.5 | 0.017 | Short process with document issues |
| 1 | 2 | 3.31 | 12.0 | 12.0 | 0.301 | Full successful applications |
| 2 | 1 | 0.01 | 3.0 | 3.0 | 0.003 | Abandoned application |
| 3 | 1 | 1.13 | 11.0 | 10.0 | 0.113 | Rejected after document resubmission |
| 4 | 1 | 2.17 | 10.0 | 10.0 | 0.241 | Rejected after exam failure |
| 5 | 1 | 3.13 | 12.0 | 12.0 | 0.284 | Successful application (variant) |

**Behavioral Pattern Analysis**

**Identified Process Patterns:**

1. **Successful Applications** (Clusters 1 & 5): Complete process flow with longest duration (~3.1-3.3 days)
2. **Early Rejections** (Cluster 0): Short-duration applications rejected due to document/criteria issues
3. **Mid-Process Rejections** (Clusters 3 & 4): Applications rejected after resubmission or exam stages
4. **Incomplete Processes** (Cluster 2): Early abandonment by applicants

**6. Process Performance Insights**

**Completion Times:**

* Successful applications: ~3 days average completion time
* Early rejections: Several hours to resolution
* Mid-process rejections: 1-2 days processing time

**Critical Process Phases:**

* **Document Verification**: Major filtering stage affecting multiple clusters
* **Examination Phase**: Significant bottleneck for candidate progression
* **Process Complexity**: Successful applications show highest activity transitions

**Bottleneck Identification**

**Primary Bottlenecks:**

1. Document submission and verification phase
2. Examination preparation and execution
3. Appeal processing workflows
4. Communication delays between phases

A table of process analysis

AI-generated content may be incorrect.

**7. Process Improvement Recommendations**

**Immediate Improvements**

1. **Document Submission Guidelines**: Enhance clarity of requirements to reduce early rejections
2. **Abandoned Application Prevention**: Investigate drop-off reasons and implement retention measures
3. **Exam Preparation Resources**: Provide better preparation materials to reduce failure rates
4. **Process Timeline Optimization**: Streamline activity intervals for improved efficiency

**Long-term Optimization Strategies**

1. **Automated Document Verification**: Implement intelligent document checking systems
2. **Predictive Analytics**: Use clustering insights to predict application outcomes
3. **Process Personalization**: Tailor process flows based on applicant behavioral patterns
4. **Real-time Monitoring**: Establish KPI dashboards for continuous process monitoring

**8. Conclusion**

Through this project, I have successfully analyzed the DAP portal application process using various process mining techniques that we learned in class. The combination of BPMN modeling, Petri nets, process discovery algorithms, and K-means clustering has helped me understand how the application process works and where improvements can be made.

During this analysis, I was able to discover the process model using three different algorithms with varying results. The Heuristic Miner performed best with perfect fitness (1.0000) and high precision (0.9785), while the Alpha Miner showed balanced performance (0.9006 fitness, 0.9759 precision), and the Inductive Miner achieved perfect fitness but lower precision (0.7845). The K-means clustering revealed six different patterns in how applicants go through the system, which was really interesting to see. I also identified several bottlenecks in the process and came up with practical suggestions for improvement.

From my analysis, I learned that the DAP portal process works well overall, but there are some areas where applicants face difficulties. The document verification phase seems to cause problems for many applicants, and the examination phase is another major hurdle. Understanding these patterns could help the DAP improve their system to make it more user-friendly.

Working with the Alpha Miner, Inductive Miner, and Heuristic Miner algorithms taught me a lot about process discovery. The results showed interesting differences between the algorithms the Heuristic Miner gave the best overall performance, while each algorithm had different strengths in terms of fitness versus precision. This taught me that algorithm selection depends on what aspects of the process model are most important. The clustering analysis was particularly valuable because it showed me how different types of applications follow different paths through the system.

One limitation of my work is that I had to create simulated data instead of using real DAP portal logs. With only 8 cases, the analysis might not capture all the variations that happen in real life. However, I believe the methodology I used could easily be applied to larger, real datasets.

In conclusion, this project has given me practical experience with process mining tools and techniques while analyzing a real-world government process. The results show that data-driven analysis can provide valuable insights for process improvement, even in complex administrative systems like the DAP portal.