МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

(НИЯУ МИФИ)

Высшая инжиниринговая школа

Отчет по Лабораторной Работе 2

По курсу «Нейронные сети и искусственный интеллект. Машинное глубокое обучение.»

Работу выполнили студенты группы М22-Ш04:

Волошин Тарас

Горбунов Максим

Москва 2023

**BPI Challenge 2015**

Сравнительный анализ пяти голландских муниципалитетов

Методы интеллектуального анализа процессов

В рамках ежегодной конференции в области управления бизнес-процессами организуется Международный семинар по анализу бизнес-процессов (BPI’15).

Организаторы этого семинара также организуют международный конкурс: the Business Process Intelligence Challenge 2015. В рамках этого конкурса, проводимого в 2015 году, участникам предлагается кейс о процедурах выдачи экологических разрешений в муниципалитетах Нидерландов. Анализ, приведенный в этой статье, был основан на данных пяти муниципалитетов, которые были предоставлены для участия в этом конкурсе. Все действия в муниципальных учреждениях регистрируются с помощью компьютерной системы. Эти данные журнала были опубликованы в открытом доступе для участия в конкурсе

**Введение:** В данной статье анализируется процесс подачи заявок на получение разрешений в пяти муниципалитетах Нидерландов с использованием “ process mining”. Анализ выявляет ряд

направлений для возможного улучшения технологического процесса и его качества, а также определяет вопросы для дальнейшего исследования соответствия процессов. Анализ также выявляет различия в организационной структуре муниципалитеты и риски для жизнеспособности небольших офисов.

**Практика**

Находим число событий и кейсов:

print(f"Число событий: {len(log)}")

print(f"Число кейсов: {len(log['case:concept:name'].unique())}")

Число событий: 262628

Число кейсов: 5647

import matplotlib.pyplot as plt

bp = log.groupby(['muni'])['case:concept:name'].nunique().reset\_index()

sns.barplot(x=bp['muni'], y=bp['case:concept:name']);

Изображение выглядит как текст, снимок экрана, Прямоугольник, Красочность

Автоматически созданное описание

mean\_length = log.groupby(['muni', 'case:concept:name'])['caselength'].mean().groupby('muni').mean().reset\_index()

sns.barplot(x=mean\_length['muni'], y=mean\_length['caselength'].dt.seconds / 86400 + mean\_length['caselength'].dt.days);

Изображение выглядит как текст, Прямоугольник, снимок экрана, диаграмма

Автоматически созданное описание

from pm4py.algo.filtering.log.variants import variants\_filter

from pm4py.statistics.traces.generic.log import case\_statistics

variants = variants\_filter.get\_variants(event\_log)

print(f"У нас есть: {len(variants)} различных вариантов путей в нашем журнале событий")

У нас есть: 5539 различных вариантов путей в нашем журнале событий

min\_count = log.groupby("case:concept:name").size().min()

max\_count = log.groupby("case:concept:name").size().max()

print(f"Минимальное количество событий: {min\_count}, максимальное: {max\_count}")

Минимальное количество событий: 1, максимальное: 154

min\_idx = log['caselength'].idxmin()

max\_idx = log['caselength'].idxmax()

print(f"Меньше всего времени: {log.loc[min\_idx]['case:concept:name']} - {log.loc[min\_idx]['caselength']}")

print(f"Больше всего времени: {log.loc[max\_idx]['case:concept:name']} - {log.loc[max\_idx]['caselength']}")

Меньше всего времени: 21573264 - 0 days 00:00:00

Больше всего времени: 3198296 - 1512 days 00:00:00

durations = pd.Series(dtype = object)

for case\_id in log['case:concept:name'].unique():

  timestamps = log[log['case:concept:name'] == case\_id]['time:timestamp']

  dt = (timestamps[1:].reset\_index() - timestamps[:-1].reset\_index())['time:timestamp']

  durations = pd.concat((durations, dt, pd.Series(pd.NaT)))

log['duration'] = durations.reset\_index(drop=True)

print(f"Событие, занимающее в среднем дольше всего: {log.groupby('concept:name')['duration'].mean().idxmax()}")

Событие, занимающее в среднем дольше всего:01\_BB\_680

Вузуализация данных по муниципалитетам:

from pm4py.algo.evaluation.simplicity import algorithm as simplicity\_evaluator

from pm4py.algo.evaluation.generalization import algorithm as generalization\_evaluator

from pm4py.visualization.petri\_net import visualizer as pn\_visualizer

gviz = pn\_visualizer.apply(net, im, fm,

                           parameters={pn\_visualizer.Variants.FREQUENCY.value.Parameters.FORMAT: "png"},

                           variant=pn\_visualizer.Variants.FREQUENCY,

                           log=log)

# metrics['inductive'] = [

#   pm4py.fitness\_alignments(log, net, im, fm)['average\_trace\_fitness'],

#   pm4py.precision\_alignments(log, net, im, fm),

#   generalization\_evaluator.apply(log, net, im, fm),

#   simplicity\_evaluator.apply(net)

# ]

pn\_visualizer.view(gviz)

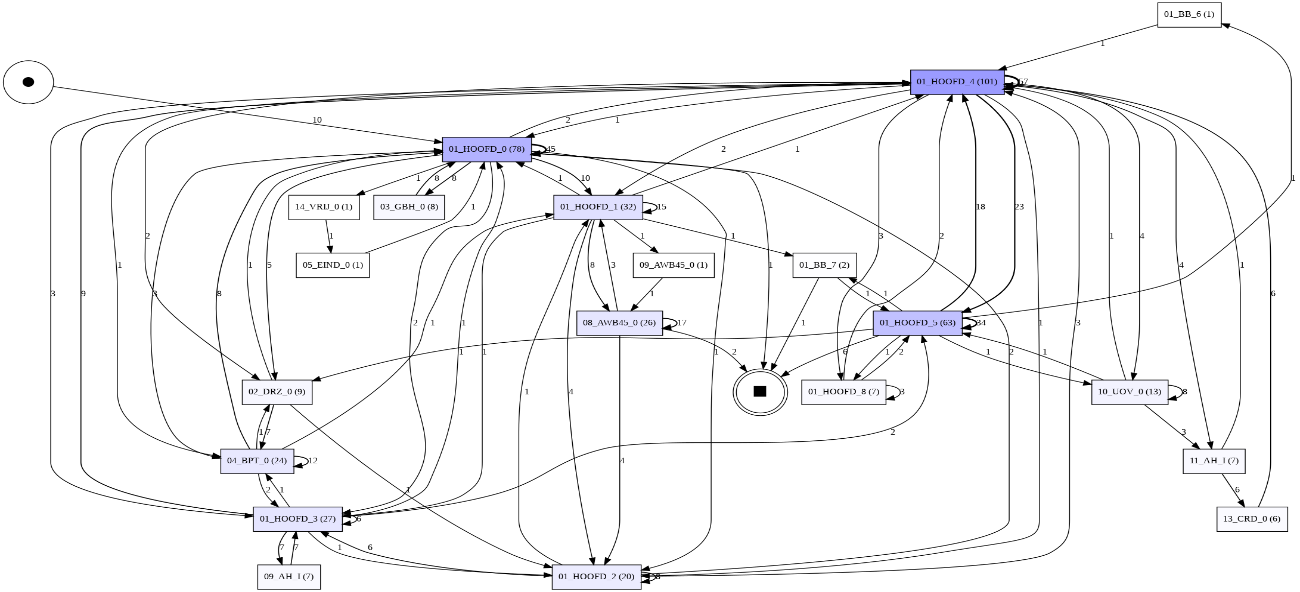
Индуктивный майнер – сети Петри

Изображение выглядит как текст, снимок экрана, линия, Шрифт

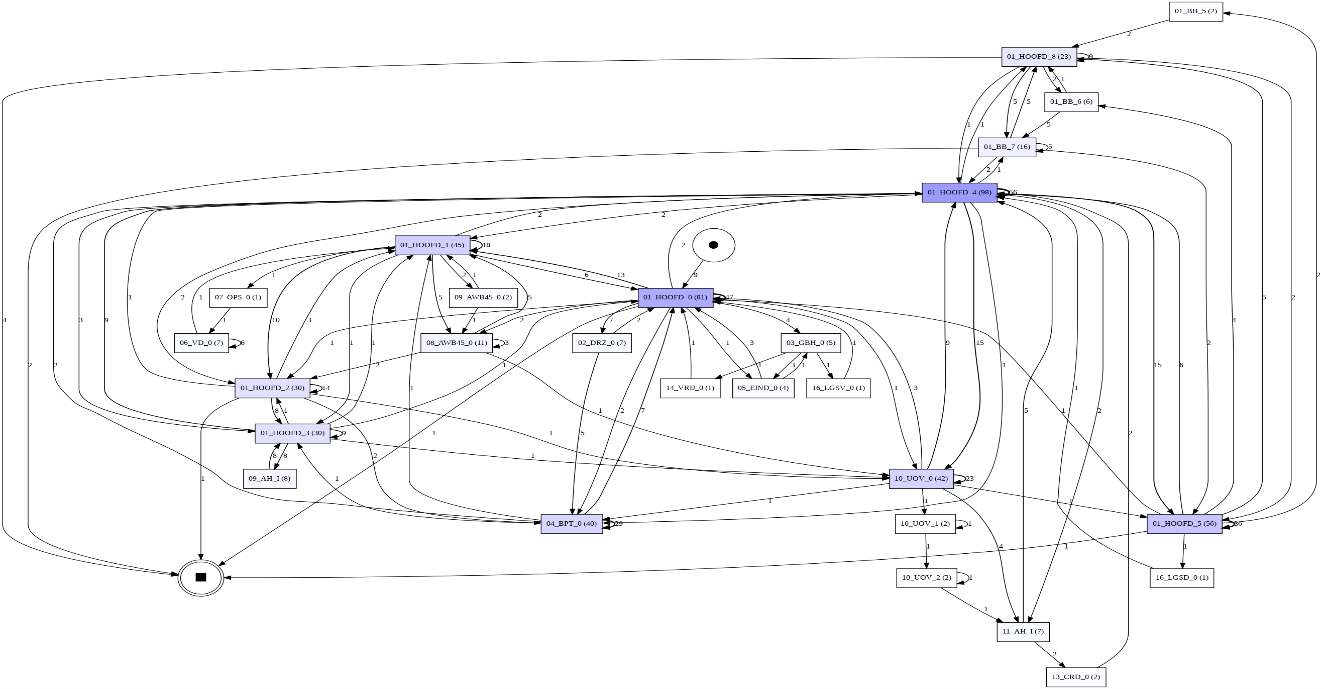
Автоматически созданное описание

Directly-Follows Graphs

События связанные с разрешениями по пожарной безобасности для муниципалитета C



События связанные с разрешениями по пожарной безобасности для муниципалитета B



Были рассчитаны количества сотрудников по муниципалитетам, больше всего нагрузки приходится на муниципалитеты B и D. В этих муниципалитетах меньше всего сотрудников, из-за чего большая нагрузка приводит к длительной обработке заявок.



Наиболее загруженные сотрудники в муниципалитетах по кол-ву дней:

