Event Log Sampling for Predictive Monitoring

Project Initiation

Documentation

Recommended by

Madhavi Shankar
Supervisor

Applied by

Date: 15.10.2021

Team 2
Malek Alhelwany, Taekeun Jeong, Xiaoyan Jin
Master students

Table of Contents

1 Overview

The event log sampling for Predictive Monitoring (LSPM) proposes an instance selection procedure that allows sampling training process instances for prediction models. We show that our sampling method allows for a significant increase of training speed for next activity prediction methods while maintaining reliable levels of prediction accuracy.

1.1 Motivation

In this project, we will implement one of these sampling algorithms developed by the chair of Process and Data Science(PADS) at RWTH Aachen University in the form of web services, combine it with existing predictive models, and produce prediction results.

1.2 Acceptance Criteria

1. The web service returns the sampled event log in CSV format.
2. with the sampled dataset, the predictive model will be trained and produces result as the one from the original implementation

1.3 Keywords

Key	Description
Web service	Web service is a software that is designed to perform a certain set of tasks. This can be any software, application, or cloud technology that provides standardized web protocols(HTTP or HTTPS) to interoperate, communicate, and data messaging throughout the internet.
CSV	A Comma Separated Value(CSV) file is a text file that has a specific format which allows data to be saved in a table structured format. Every row contains values of different attributes(column), which are separated by comma.
XES	eXtensible Event Stream: standard is to standardize a language to transport, store, and exchange (possibly huge) event data (e.g., for process mining)
LSPM	Event Log Sampling for Predictive Monitoring : Instance selection procedure that allows sampling training process instances for prediction models, of which accuracy is similar to the result of the model using the whole event log.
Predictive Monitoring	Subfield of process mining that aims to estimate case or event features for running process instances such that stakeholders can, for example, predict undesired to prepare and minimize risks.

Table 1 Keywords used in the study

2 Business Use Case

In this part, we will discuss the business use case of the tool and the need to employ such one in big projects of process mining where the big data is always a challenge when we need to train machine learning models in order to predict future activities.

2.1 Scope

We created the Log Sampling for Predictive Monitoring (LSPM) tool, to make the use of machine learning algorithms in the field of process mining much more efficient, faster and productive. It will help the user to train the ML model with sampled event log much faster and with less usage of memory, while maintaining accepted results of prediction accuracy. The input will be any event log of CSV/XES extension, with focus on many characteristics within the original event log:

- **Many cases and activities with few variants.**
- **Many cases and activities with many variants.**
- **Few cases and activities with more variants.**

And according to this characteristics, we employed different settings of sampling parameters for our tool:

- **Unique selection**
- **Logarithmic distribution**
- **Division**

2.2 Key Benefits

When we want to train the machine learning model, selecting the right method is a key, but then using the LSPM tool is also another important key when it comes to complex or big event logs.

For example: using division technique, then all variants in the event log have at least one trace in the resulting sampled event log, but way smaller than the original event log while keeping the same key points that leads to a good trained model with a good accuracy.

Using LSPM plugin will:

- **Save time needed to train big event logs.**
- **Save more memory space when dealing with sampled event log (to use for training).**
- **Increase the speed of computing the training model.**

So our main goal is to make things faster, easier and clearer for the end user when it comes to train the machine learning model with complicated or big event logs in order to achieve a good accuracy for predictive monitoring tasks.

3 Feasibility Study

We are checking the project feasibility from a theoretical and technical point of view, and we pointed out all risk points which affect our project, in order to prevent them and have a useful tool, which is able to integrate to any process mining project.

3.1 Theoretical View

We discuss here a high-level point of view, how we will implement our tool, which strategies we use and what are the non-technical requirements we have to achieve.

We will develop a web service that accepts user input event log data, samples instances, and returns training sample event logs in CSV format.

We will mainly focus on sampling the representative traces from each variant. In other words, we will extract traces from each variant, for example, the most frequently occurring attribute values, such as most frequent resources.

At first, we will determine every variant and their corresponding traces. Second, the prioritizing values, such as the frequency or mean values of their attribute values, are measured. Third, every trace is sorted based on these values. After that, the number of extracted traces is computed. Finally, based on this number, the traces are returned as a sampled dataset. This number can vary based on different settings. The following describes different schemes.

1. Unique selection: there is just only the trace with the most frequent attribute value is returned, in other words, there will be only one trace of each variant is returned
2. Logarithmic distribution: Let N is the number of whole traces from each variant, there will be log k[N] traces returned based on the given k
3. Division: Let N is the number of whole traces from each variant, there will be [N/k] traces returned based on the given k

3.2 Technical View

For the technical view, we will use the following tech-stack for our project:

Python (v3.10.0): A general-purpose, versatile and powerful programming language, also one of the most popular programming languages for Data Science and Process Mining.

Django (v3.2.8): A high-level Python web framework that encourages rapid development and clean, pragmatic design.

Pm4py (v2.2.15): Python library used for process mining algorithms.

Docker (v20.10.9): An open source containerization platform. It can help the developer to pack and run the application in a loosely isolated environment. Its highly secured and isolated therefore can allow us to run many containers simultaneously.

4 Risks

We identify the following risks and provide corresponding mitigation strategies, so we can react properly and prevent our project from negative consequences.

Risk	Mitigation strategy
Running out of time	Focus on the main deliverables contributing to the project's progress.
Underestimation of task's complexity	Allocate more team members to work together.
Technical difficulties	Clarify difficulties with team members or look for external help.
Team member drops out	Meet to reallocate responsibilities.

Table 2 Risks and mitigation strategies

5 Project Plan

5.1 Gantt chart!

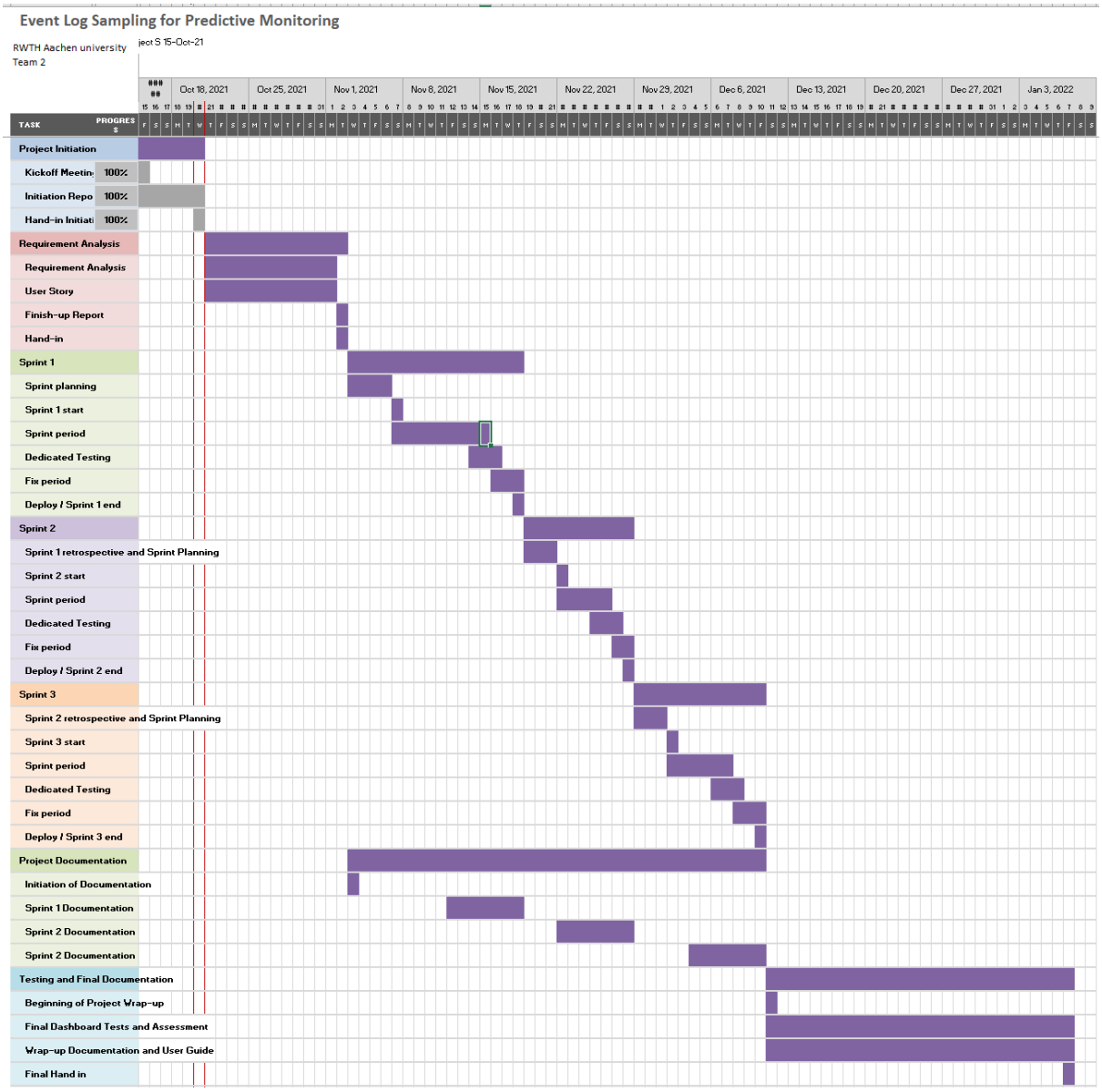

Table 3 Gantt chart

6 Project Team

Malek Alhelwany:

A master of computer student, with an AI background, interested in the field of data science, has a good experience developing web and mobile applications.

Key responsibilities: Pm4Py, Python, Code Structure, Django, Requirement Analysis

Taekeun Jeong:

A master of data science student, interested in the field of statistics, machine learning, and process mining, has a good experience in GUI-Development based on python and C++.

Key responsibilities: Docker, Pm4Py, Ui

Xiaoyan Jin:

A master of data science student, interested in the field of machine learning, process mining and has good experience with python and R.

Key responsibilities: Unit test, Requirement Analysis, Django, Python