Project Name	*Contrastive Learning for Event Sequences with Self-Supervision on multiple domains*
Project Type	*Replication/Research*
Team Name	*ClosedAI*
Project Team	1. *Denis Grankin – took online ML courses, has basic understanding of DL (no practical experience), has a little experience in practical ML* 2. *Ekaterina Andreichuk - newbie to Deep Learning, first time took ml course* 3. *Mikhail Konenkov - new to DL, first-timer in ML courses, read a lot of articles on ML and DL algorithms and problems* 4. *Ivan Apanasevich - newbie to Deep Learning, first time took ml course* 5. *Irena Gureeva - newbie to Deep Learning, have a little experience in ML*
Who suggested	*Maria Ivanova (TA)*
Difficulty	*Easy/Medium*
Proposal	*Pytorch-Lifestream (Ptls) library https://github.com/dllllb/pytorch-lifestream designed for generating embeddings from complex multimodal data of various event sequences types: clickstream, customer purchases, telecom phone calls, banking transactions and etc. Embeddings can be generated by several methods and then can be used in the subsequent downstream machine learning tasks. In https://arxiv.org/pdf/2002.08232.pdf, the authors proposed a novel method CoLES to generate event sequence embeddings in self-supervised scenario with contrastive loss. In this project, we are going to understand the CoLES paper, get familiar with the Ptls library, replicate the several methods from Ptls (agg baseline, random encoder, coles) for the two types of data (transactions and clickstream) and compare its results on downstream task.* *Main goals are to understand the paper, replicate the mentioned methods on different types of event sequence data and compare the results.*
Specific Tasks & Expected results	1. *Get familiar with the ptls library https://github.com/dllllb/pytorch-lifestream* 2. *Take dataset with transactions and clickstream for experiments (https://data-fusion.ru/data-fusion-2022/sorevnovaniya/).* 3. *Run the proposed methods several times for learning embeddings on two types of data (transactions and clickstream).* 4. *Implement downstream machine learning task on both types of embeddings and compare its results*
Relevant papers	• *https://arxiv.org/pdf/2002.08232.pdf* • *https://arxiv.org/pdf/2009.00104.pdf*
Grading Scheme	*90% Describe the algorithm, train the proposed models in self-supervised scenario and downstream task, calculate relevant metrics, compare the results;* *70% Describe the algorithm, train the proposed models in self-supervised scenario only*