ALEXANDER NIKOLIN

Saint-Petersburg, Russia @ apnikulin@edu.hse.ru linkedin.com/in/howuhh github.com/howuhh

EXPERIENCE

Machine Learning Engineer Intern

CoreML, V Kontakte LLC

July 2019 – Sep 2019 Saint-Petersburg

- the purpose of the internship was to solve a link recommendation problem. I wrote a pipeline for graph embeddings training using matrix factorization (ALS) and top-n recommendations of the most similar users using distributed kNN search.
- for the correct sampling of the graph and the hyperparameters tuning, I also wrote my own Metropolis-Hastings random walk implementation on pyspark. The results were tested in online using an A/B test.

Teaching Assistant

Higher School of Economics

Sep 2017 – May 2018 Saint-Petersburg

- my responsibilities were checking homework and tests, answering students questions, and helping with practice assignments on seminars, mostly R and SQL

HACKATHONS

Gazprom Neft - Client Clustering

Ai.Hack by Innopraktika

2018 Saint-Petersburg

- I was involved in clustering embeddings derived from client transaction sequences using word2vec and then analyzing and visualizing the resulting clusters (using t-SNE)

Coursera courses recommendation by Stack Overflow profile

Data Science Hackathon EPAM

2017 Saint-Petersburg

- I collected all course descriptions from Coursera, clean up the data, tokenize and lemmatize the words, prepare a final dataset for a content based recommendation system based on the term vector model and cosine similarity

EDUCATION / COURSES

Sociology and Social Informatics

Higher School of Economics

Sep 2016 – May 2020 Saint-Petersburg

Open Machine Learning Course

OpenDataScience

October 2018 – December 2018

43 out of top 100 (from 1k+) on final rating

PROJECTS

End-To-End Churn Prediction Report

- detailed exploratory analysis of churn structure, data preprocessing and selection of important features
- model construction and validation for churn prediction, analysis of specific cases of algorithm prediction using LIME, calculation of possible profit

Single-Linkage Clustering

- my implementation of agglomerative single linkage clustering with minimum spanning tree algorithm from "Modern hierarchical, agglomerative clustering algorithms" paper by Daniel Müllner.

Graph Nodes Similarities

- module which allows you to calculate similarity metrics based on local information for the nodes of the graph, ranking by the number of the most similar ones, and easily add new metrics.
- this implementation can also be used to process large graphs, as all calculations are written in PySpark.
- all metrics are taken from the "Predicting Missing Links via Local Information" paper.

Metropolis-Hastings Random Walk

- module, as well as CLI to it, which can be used to sample unbiased samples of large graphs that are too difficult to process completely or are not suitable for tuning hyperparameters of algorithms on graphs due to too long processing time.
- parallelism and distributedness of the algorithm is achieved with the help of PySpark.

SKILLS

Languages: Python, R , SQL

Data Science Stack: pandas, numpy, scikit-learn, scipy, matplotlib, hyperopt, catboost, pyspark, lime

Tools: Git, Tmux, Tableau, Pyenv