Nikolai Golikov

Python software engineer

Birth date: October 4, 1969

Techs/Skills: Programming languages:
Advanced level: Python, pure C.
Other: JavaScript, Swift, Lua, Scheme
IDE: Vim

Preferable OS: Linux (Debian)

Python web/networking/database frameworks/libraries:
Django, Flask, SQLAlchemy, Requests, Mechanize, Scrapy, LXML, PyMongo
Python scientific/data science frameworks/libraries:
NumPy, Pandas, SckitLearn, Dask, Matplotlib, SciPy

JavaScript frameworks/libraries:
jQuery, AngularJS

Databases:
SQLite, PostgreSQL, MongoDB

Networking protocols/formats/architecture:
HTTP, REST, JSON, XML, XPath

Practical experience:

June 2016 – present, Salford Systems

CloudSML Computational backend.
1. Partly implemented distributed version Gradient Boosting Machine (by Dask and Numpy)
2. Implemented components for data processing, model building, and scoring results (by using Dask, Pandas, NumPy, PFA, Scikit-Learn).

November 2015 – June 2016.
Work as freelancer.
1. Many complex scrapers.
Used lang/technologies: Python, Scrapy, Python Requests, SQLAlchemy.

2. Web app, that aggregates worldwide universities rankings: QS - QS World University Rankings, THE - Times Higher Education, ARWU - Academic Ranking of World Universities, NTU - National Taiwan University Ranking, URAP - University Ranking by Academic Performance. Scraping data from each ranking site. Matching univiersities names from different rankings. Buiding aggregate ranking based on selected by user rankings (or all). Calculating various stats

(correlation matrix, etc.).
Used lang/technologies: Python, Django, Django REST Framework, Requests, LXML, Pandas, JavaScript, AngularJS

3. Some scripts for Kaggle competitions.
Used lang/technologies: Python, Pandas, ScikitLearn, Matplotlib

4. Some unsupervised machine learning problems (clustering statistical data) with results visualisation.
Used lang/technologies: Python, Pandas, ScikitLearn, Matplotlib

2013 – 2015 Coderivium Kharkiv
1. Web Labirynth game.
Give, by send user (bot) REST request, labirynth (maze) node with conjugate nodes various type. User must find minimal path, or all possibility minimal paths, from begin to end node passing through all special (“power pill”) nodes. Labirynths generates automatically, by algorithm that create arbitrary specified size labirynth, with specified size complicated (such that admit many pass through nod) minimal path with specified number special nodes. Algorithm developed specially for this application.
Used lang/technologies: Python, Flask, SQLAlchemy, NetworkX, REST API, JSON

2. Device enrollment system
System integrates with Apple Developer Member Center, build distribution system and buildserver. This allows developers to enroll customer's devices in Apple portal and automatically update digital signatures on buildserver and frees developer from dealing with device identifiers and signature files
Used lang/technologies: Python, Flask, SQLAlchemy, Jinja2, Mechanize, Celery, Python-LDAP, JSON, LDAP, jQuery, Javascript, REST API, Twitter Bootstrap

3. Some simply and complex scrapers.
Used lang/technologies: Python, Python Requests, Python multiprocessing.

4. Optimization the smooth rendering of color changes by moving from RGB color space to CMYK and rotating in this space
Used lang/technologies GL Shaders, Unity3d C#, Objective C.

2001 – 2013
During this time I worked as a system administrator (Supporting a mixed computer net, various versions of Windows + Linux, Samba, Remote Desktop, Iptables etc.).

Also I completed several one-off orders requiring knowledge in the field of mathematics or mathematical programming:

1. An analytical solution to the problem of the oscillations of the axle-wheel system when the car turns. The problem was reduced to solving a fourth-order partial differential equation. The solution was obtained using a modified Fourier method («Theory of Elasticity» by S. Timoshenko and J.N. Goodier). It solution does not describe the real situation very well. More suitable numerical or approximate methods.

2. Program allowing to look for singular points (and to determine heir type) in the parametrized systems of differential equations (Lotka–Volterra equations’ generalization) describing some social

processes.
Used lang/technologies: Python, NymPy, SciPy, OpenOpt.

Education:

Specialist (Master) in applied mathematics
Educational institution issuing the diploma: Donetsk National University
Year of graduation from the university: 1997
Specialization: mathematical problems of the theory of elasticity

Self-education after graduating from the university:

In order to support the skills of «mathematical» or rather «nature-science» thinking I study (if there is time) the literature related to the problems I am interested with.
- in 2006-2007 I read «Structure and Interpretation of Computer Programs» by Abelson and Sussman, solving almost all problems from Chapters 1, 2, 3 and partially from Chapter 4.
- in 2010 I read several chapters of «A first Course in General Relativity» by B. Shutz (along with solving the problems given there).
- from time to time I read and re-read «Introduction to algorithms.» by Cormen, Leiserson et al. (along with solving the problems given there).

Periodically I re-read the «Feynman lectures on physics.» and solving problems.
I also decided study again the basics of probability theory and mathematical statistics. Currently I'm studying «Introduction in probability theory» by A. Kolmogorov et al.

I have also completed some courses on Coursera (2013 year):
"Algorithms: Design and Analysis, Part 1" (score 82.7/100), "Machine Learning" by Andrew Ng (score 100/100), "Exploring Quantum Physics" (score 90.6/100 with distinction).

Some info about myself:

In my childhood I was fond of mathematics, physics, astronomy, I read a lot of popular science literature. I learned many things that go beyond the school curriculum. For example, I learned how to integrate and differentiate independently, at the age of 12 years using Vygodsky's guide to higher mathematics. Much of what I still remember, I learned before entering university.