Boris Dev

boris.dev@gmail.com

Interests

I get data projects unstuck. I start by using cognitive empathy to get my head around the mental model of the end user (CEO, developer, data scientist, customer, SME, etc), and then work backwards, searching for the "low-hanging fruit" changes. Often this optimization involves uncovering extraneous cognitive load and wrong assumptions that build up in the split between two siloed teams. My toolkit to continuously question assumptions, and experiment, includes agile code refactoring, demos, papers (examples), and building observability systems.

- Process optimization and system design: domain modeling, debugging hand-off toil, uncovering extraneous cognitive load, building observability systems (monitoring and KPIs)
- Data Analysis: exploratory data analysis, causal inference
- Token economics: Ethereum, token curated registries
- NLP AI: fine-tuning large language models (GPT-3)

Coding tools

- Product developer: Linux, Python, Postgres, Flask, Django
- Infrastructure: Docker Compose, Kubectl, Kafka, Celery, Helm, LightStep, ELK, Splunk, AWS Sagemaker, AWS Lambda, Grafana, Prometheus
- Data science: Pandas, Numpy, PyTorch, Tableau, AWS Groundtruth

Job experience

Technical AI Product Manager consultant at SimpleLegal, 2022 - 2023

The company's first AI feature was stuck. As tech lead, my puzzle was to figure out why the performance of their machine learning models for text classification was flat even after the company had been spending more money on human annotation. The actions I took resulted in the company launching their first AI feature, with positive feedback from customers and the sales team. Below are the sequence of actions that I took.

- I stopped the human annotation process (pulled the Andon Cord).
- I identified an incorrect assumption: our performance blocker was not training data quantity, but quality. Then I identified two main culprits of the poor training data quality: 1) convoluted annotation guidelines and 2) missing pre-processing noise filters.
- I immersed myself in the metrics to perform triage on eleven machine learning classifier models: one was never needed; two were replaced by expert rules; five were sufficient; and two were targeted for relabeling.

- I immersed myself in the company's legal invoice data through exploratory data analysis and by labeling several thousand sentences while continuously getting feedback from our subject matter expert (SME).
- I simplified the annotation guidelines in collaboration with both the SME and annotation team.
- I designed a new annotation Human-in-the-loop ML QA process in collaboration with the SME and annotation team. This included a CI (continuous improvement) process where the annotators, our SME, and myself reached consensus to fix the guidelines as we hit edge cases (ie. feedback).
- I added pre-processing noise filters to the labeling pipeline (AWS GroundTruth).
- I wrote papers to explain new concepts and changes for the product team and executives.
- I worked daily with our NLP-ML expert on re-prioritization of R&D work (ie. triage).
- I added a new QC process (embarrassment review sheets and staging server).
- I refactored the inference server (AWS Sagemaker) with new post-processing, decoupling, thresholds and preprocessing noise filters.

Backend developer at Sight Machine, 2018 - 2021

The puzzle I had was to bring the company's biggest public facing feature at the moment, Recipes, from its embryonic start as a spreadsheet to general release. Beyond the standard work required as the lead backend engineer (query optimizations, analytic endpoint coding, etc), my critical non-technical effort included release coordination (triage) and non-technical explanations to the product and customer support teams on what was going on "under the hood", and continually incorporating their feedback.

In addition, I made analytic endpoints (Flask, Numpy, Pandas, Celery, Sqlalchemy) and wrote high level design papers. I started the company's first distributed tracing using LightStep. I simplified our development environment. And, I occasionally wore a devops hat by using Kubectl, Helm Charts, and Grafana.

Data and product engineer tech lead at HiQ Labs, 2015 - 2018

My company provided predictions to our customers on whether their employees were about to quit. As tech lead, the puzzle for the CTO and I was to figure out how to get around LinkedIn's bot detection in order to scrape millions of HTML public profiles, the raw data for our prediction pipeline. To solve the puzzle I ran and tracked experiments on different spider configurations.

Another puzzle we had was to reduce the stressful, manual pipeline work involved in every monthly release. I led developers and data scientists to move from a monolithic pipeline to a microservice pipeline (Spark, Kafka, Rancher, Docker

Compose). A major effort was spent in refactoring old code so it could be decoupled into separate services, achieving horizontal scaling, reduced tech debt and reduced cognitive load during releases. In addition, I designed and built the company's first observability system, and I trained data scientists on Kafka and Spark, and junior developers on coding.

Start-up partner and developer at Map Decisions, 2014

I created a mobile app to automate street sign inspection (Angular, Django)

Developer at Urban Mapping, 2011 - 2013

We provided a location query and map tiling service to Tableau's software. The puzzle I had was to stop embarrassing regression errors. Regression errors occur when a developer's bug fix breaks something that had previously worked. I identified the culprit. Our developers found it too complicated to deduce analytically the impact of their bug fixes because of the very large variation in customer requests (combinatorial explosion from map rendering of hundreds of variables at a dozen scales). As an empirical solution, using clustering and histograms, sorted by frequency and latency, I formed two samples of representative test requests: 1) two hundred test requests were automatically run after every git push and 2) a dozen test requests were manually run locally by developers doing TDD. In addition, I designed and built the company's first observability system (Splunk and Tableau) and created the company's first Jenkins QA CI system.

Open source and work papers

- Co-founder of library for clustering geographic areas, github.com/clusterpy.
- A play Ethereum MEV bot, github.com/borisdev/play_mev_bot
- A git bare approach to version control your dot files, github.com/borisdev/dotfiles
- Work papers

Academics

- Ph.D dissertation: Assessing Inequality using Geographic Income Distributions 2014.
- Entry in Encyclopedia of Human Geography on Spatial Econometrics. Sage Publications. 2009
- Interactive spatiotemporal modelling of health systems: the SEKS–GUI framework
- sigma-convergence in the presence of spatial effects
- Integrating Econometric and Input-Output Models in a Multiregional Context