

Boris Dev

boris.dev@gmail.com

Interests

- Building observability systems
- Synthesizing and documenting diverse expertise into a common language
- Process optimization to reduce toil

Tools

- **Backend engineering:** Linux, Python, Postgres, Flask, Django
- **DevOps:** Docker, Jenkins, Helm Charts, AWS Sagemaker and Lambda
- **Observability:** Splunk, Lightstep for distributed tracing
- **Data science:** SQLAlchemy, Spark, Pandas, Numpy, PyTorch, Tableau
- Inviting feedback with demos and papers ([examples](#)).

Job experience

Data Scientist / Technical AI Product Manager, consultant at SimpleLegal, 2022 - 2023

The SimpleLegal product is a tool for legal departments of large companies to manage the invoice bills from their vendor law firms. Before I arrived, the company had been working on a language AI feature for flagging suspicious line items. For one year, this feature was stuck because of bad performance. They were unable to increase performance even after spending more money on human annotation. My job as tech lead was to increase AI performance. The actions I took resulted in the company launching their first AI feature, with positive feedback from customers, the sales team, as well as an AI Excellence Award. Below are the actions that I took.

I managed the data science consultant.

- To increase clarity, I helped the data scientist by organizing his analysis of the machine learning metrics into a table, which was continually shared at the bi-weekly stand-ups to help others comprehend our progress.
- In the spirit of shared success, I worked daily as a team member with the data scientist doing exploratory data analysis of the invoice data. This resulted in the following triage: We killed one classifier that was never needed, replaced another with expert rules, stopped the labeling on five whose performance was sufficient, and lastly, narrowed the focus of the labeling effort from eleven to two classifiers. In addition, we identified two culprits of bad quality labeling, 1) convoluted annotation guidelines and 2) missing pre-processing noise filters.
- I asked the data scientist to share him alternative explanations of complex concepts to the rest of the team to help justify our changes.

- I wrote him a stellar performance review.
- I acted decisively to pause the human annotation process until after we made the required fixes.

I managed the offshore annotators.

- To understand the perspective of the offshore annotators, I labeled several thousand sentences.
- I built an external relationship with the offshore team of annotators. My objective was to increase training data, or labeling, quality. I managed them with a spirit of continuous improvement and consensus. I asked them to debate the subject-matter expert and I over Slack when they disagreed with us.

I built relationships across the organization.

- To teach me, I connected with experts spread across three sisters companies. This group included a principle data scientist, two product subject-matter experts, one of which was a founder and VP, two DevOp experts, and one human annotation expert.
- To seek clarity, at my request the president convened a team meeting in the style of the Amazon 6-pager. The team spent the first fifteen minutes reading my high-level design document explaining that the performance blocker was not training data quantity, but quality, as well as other clarifications.
- I assigned Jira issues to the engineers and data scientist.
- After the engineer helping me left to another job, I stepped in to refactor the inference server (AWS Sagemaker) with new post-processing, decoupling, thresholds and preprocessing noise filters.

Backend developer at Sight Machine, 2018 - 2021

The Sight Machine product is a dashboard tool for process engineers at manufacturing plants to keep track of performance.

- I built relationships across silos (customer success, devops, data engineers, and product engineers) in order to bring the company's biggest public facing feature at the moment, [Recipes](#) from its embryonic start as a spreadsheet to general release. Using high-level design papers, I gave non-technical explanations to the product and customer support teams, allowing them to get accurate feedback on product design decisions. I started a new process for giving demos to customers.
- My technical contributions included starting the company's first distributed tracing using LightStep, simplifying our development environment.
- I mentored junior engineers.

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

Our product provided predictions to our customers on whether their employees were about to quit.

Tech lead for scraping

- 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.
- I ran and tracked experiments on different spider configurations.
- I led a junior devops engineer to help me build a Splunk observability system.
- I trained a junior data engineer to help with scraping.

Tech lead for migrating from a monolith and a microservice architecture

- I refactored the pipeline code into a new repo structure.
- I trained a junior data engineering in docker-compose, Kafka, and Rancher, so he could train the data scientist in decoupling their code into docker microservice containers.
- I trained a junior data engineer in optimizing his local development environment, so he could make the data scientists more efficient.
- I explained to management the architecture using whiteboard diagrams.

Helped data scientists

- I migrated the data scientists from analysis using Mongo queries to analysis using pySpark.
- I shadowed the data scientists to see how I can help them to reduce their toil.

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

Our product 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 combination of potential parameters in Tableau customer requests. 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 using a script I made them.

- I built the company's first observability system (Splunk and Tableau).
- I built the company's first Jenkins QA CI system.

Non-software jobs

- Assistant Manager at Gundy's Grill restaurant at Beaver Creek Ski Resort, Avon, Colorado.
- Kids Snowboard instructor at Beaver Creek Ski Resort, Avon, Colorado.
- Team lead of four analysts at California Business Magazine gathering financial data on the performance of local public companies for newspapers such as the Los Angeles Times and Boston Globe.

Open source and papers

- Co-founder of library for clustering geographic areas, github.com/clusterpy. Taught students in location data, object-oriented design and UI programming.
- 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 design papers](#)
- [Why did your language AI feature fail?](#)

Academics

- Mentored students in the master degree program.
- Assisted teaching a class in spatial statistics.
- PhD 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](#)