

Python Software Engineer | Statistician

1040, Brussels, Belgium
rohail.taimour@gmail.com |
+32 489 83 64 76 | [Personal website](#) | [Linkedin](#) | [Github](#)

Education

MSc in Statistics - KU
Leuven, Leuven, Belgium -
2014-2016

Graduated Cum Laude, Master's
thesis on continuous optimization of
production processes in MATLAB

**BSc (Hons.) in
Accounting and Finance**
- Lahore University of
Management Sciences (LUMS),
Lahore, Pakistan - 2010-2014

Graduated with Distinction (3.6/4.0)
Courses: Operations Research,
Supply Chain, Decision analysis,
Applied Probability
Treasurer for University Adventure
Society organizing hiking trips for
groups of upto 300 people in North
of Pakistan

Technical competencies

Programming Languages: Python and R
with 7+ years of experience
Cloud Services: AWS (S3, EC2, ECS,
Cloudwatch, SageMaker Studio), Azure (Blob, Databricks, Pipelines)
CI/CD: Git, GitHub Actions, Azure Pipelines, GitHub CLI, Docker, Docker compose
Databases: PostgreSQL, Redis, SQLite3, Neo4j, SQLAlchemy, Alembic
Monitoring: Grafana, Sentry, Kibana
Code quality tools: pytest, coverage, responses, factoryboy, faker, pre-commit, ruff

Rohail Taimour

Summary

Seasoned Python Software Engineer with a Master's degree in Statistics. I am proficient in building applications following best software practices such as CI/CD, containerization and operating in an AWS or Azure cloud environment.

Job History

- Python Backend Engineer** at Engie GEMS, Brussels, Belgium | *Jan 2024 - Present*
- Python Software Engineer** at Illumina, Mechelen, Belgium | *April 2023 - October 2023*
- Machine Learning Engineer** at GSK Vaccines, Brussels, Belgium | *October 2022 - Feb 2023*
- Lead Data Scientist, AI Developer** at UCB BioPharma, Brussels, Belgium | *August 2016 - October 2022*

Relevant Experience

Backend Engineering for Green Energy Trading Platform using Flask and Python Ecosystem

Python Backend Engineer, Engie Green energy Management solutions (GEMS), Brussels, Belgium Jan 2024 - present

- Develop feature and incident support for business processes to trade wind and solar assets on the European electricity market, as well as managing energy contract lifecycle for Engie clients.
- Participate in off-hours guard duty to quickly diagnose and resolve critical production incidents using a combination of **Kibana** and **Sentry**.
- Fixed a long existing race condition by employing a Redis based lock.
- Replaced a legacy email notification system by simplification, modularization and rolled out the improved system using a staged roll-out with feature flags to minimize disruption of existing processes.
- Participated in and upheld a strong production-first engineering culture, where features and bug fixes are verified in production-like review environments, CI/CD using github actions, 100% test coverage on diffs, encouraging small and well tested changes to production
- Employed factories and dependency injection patterns to better represent production complexities in tests and extended our internal celery mock for verifying background task execution and SQLAlchemy session listeners behaviour.

Python solution enabling launching, pipeline management and automated report downloads for Customer-Uploaded Data

Python Software Engineer and Data Pipeline Architect, Illumina, Mechelen, Belgium April 2023 - October 2023

- Designed a Python service that automates the monitoring and processing of customer-uploaded sequencing data, initiating further analysis or report generation based on predefined criteria.
- Implemented a dual-layered approach: the first layer handles the initiation and tracking of analysis pipelines, while the second layer is registered as a **Docker** image in the analytics backend to perform post-processing on the output files and create comprehensive summary reports for the customer.
- Scheduled the Python service to operate every 30 minutes for new data and updates, ensuring seamless progression from data upload to final report delivery to customer environment.
- Implemented comprehensive systems integration, utilizing a combination of CLI tools and API calls for effective coordination and automation across various software components.
- Applied Object-Oriented Programming (OOP) principles to organize API, database interactions and endpoint processing to reduce code duplication and utilize self-documenting object names.
- Implemented unit testing using **pytest** and implemented fail-safe mechanisms for robust error handling.

Automated SQL Script Generation to facilitate PostgreSQL Data

Personal details

- Nationality: Belgian, Pakistani
- Languages: English (fluent/bilingual), Urdu (Native), French (B1)
- Mobility: Driving License available, flexible for hybrid setup in Belgium
- Availability: TBD
- Hobbies: Drumming and percussion instruments, Boulderling/Climbing, productivity, Squash, reading

Migration in multiple environments

Python Software Engineer, Illumina, Mechelen, Belgium

July-Aug 2023

- Designed and implemented a data ingestion framework to parse and validate input files for generating and validating SQL update statements.
- Conducted comprehensive testing of generated SQL scripts using a mock of production database tables to test that SQL scripts run as expected.
- Utilized SQLAlchemy for database schema management, creating and populating mock tables in a test environment to ensure the integrity and functionality of SQL scripts.
- Implemented the solution as a Python package encapsulating the entire data migration logic within a Docker entrypoint for portability and ease of deployment.
- Leveraged **Jinja2** templating to generate dynamic, parameterized SQL scripts, enabling the script to adapt seamlessly across different deployment environments, such as Development, Integration, and Production.

Yield optimization for batch and continuous production processes using Machine Learning in Python

Lead Data Scientist, Supply and Manufacturing, UCB Switzerland/Belgium Aug 2020-Oct 2022

- Production setting proposed by model directly led to an increased throughput of 20%, turning in a recurring 1.5 million euro in annual cost savings
- Analyze time series data collected from equipment sensors and visually summarize golden batch insights
- Created (Bayesian) and tree-based regression models to quantify impact of process changes and predict batch performance
- Performed a thorough model validation and hyperparameter tuning exercise before recommending model insights be tested in a live production environment
- Supported delivery of workshops demystifying the process of conducting AI projects and machine learning to process experts

Scientific influencer (KOLs) identification, ranking and profiling using network analytics and Neo4j

Data scientist, Drug Development, Commercial, Medical affairs, UCB

2018-2019

- Supported delivery of a custom Key Opinion Leader (KOL) intelligence platform based on a Python-based ETL framework to ingest and harmonize external datasets (e.g., PubMed, ClinicalTrials.gov) into Neo4j, producing custom graph databases for each project.
- For example, in an exercise to optimize the sales workforce for a given region, instead of ranking KOL's based on academic metrics, ingested additional information about prior contact with physicians into Neo4j, allowing commercial team to prioritize known physicians instead of going for an agnostic optimal solution.
- Utilized **py2neo**, neo4j web UI, and Cypher for interactive data manipulation and ools like **NetworkX** and **Cytoscape** for network visualization.

Unit Commitment Solver for Power Grid Optimization via FastAPI

- Developed a REST API using **FastAPI** for optimizing energy distribution among powerplants based on load requirements and fuel costs.
- Implemented multiple algorithms to solve the **unit-commitment problem**, considering factors like fuel cost, powerplant efficiency, and environmental constraints.
- Utilized **Pydantic** for data validation and schema definition, ensuring data integrity and streamlined request handling.
- Packaged and containerized the application using **Docker**, with detailed documentation and a Dockerfile for easy deployment and scalability.
- Employed **pytest**, along with Python best practices such as typing and linting.
- Managed project dependencies using **Poetry**, facilitating efficient workflow and package management.
- Deployed the API service using **Uvicorn** and integrated a **Swagger UI** for interactive API documentation and testing

Personal projects

Web Scraper to analyse Property Purchase and Rental Trends in Belgium

- Developed web scraper using Beautiful Soup to collect information such as apartment data such as price, area, etc.
- Implemented SQLite for data storage, using `Pydantic` for data validation and `SQLAlchemy` for database interactions.
- Encapsulated the concerns into a python package with dependency management using Poetry.
- Employed **Prefect** for job orchestration, managing the workflow's scheduling and monitoring of scraping tasks.

Personal Portfolio and blogging website built using Hugo and hosted using Github Pages

- Created website using **Hugo** and implemented features such as a contact form, and visitor commenting capabilities.
- Hosted the static website on GitHub Pages and automated the deployment process using GitHub Actions.
- Codebase hosted on [github](https://github.com)

Automated Resume Builder and Continuous Deployment System with GitHub Pages Hosting

- Engineered an automated system for generating, versioning, and hosting a dynamic CV using Markdown, HTML, Jinja templating and CSS.
- Set up a trio of GitHub repositories to separately manage the CV's content, styling, and public hosting on Github Pages.
- Developed a Python package for automating the styling and generation of the CV, integrating with Markdown and HTML/CSS.
- Implemented version control for CV content using a private GitHub repository, ensuring secure and organized data management.
- Leveraged GitHub Actions for automating the CV's generation and deployment process, enabling updates through git pushes.
- Hosted the final CV on GitHub Pages, providing a live, online version that can be easily updated