

ROBIYA OLIMJONOVA

Data Scientist | AI Engineer

Tashkent, Uzbekistan • (97) 749-3550 • [Email](#) • [LinkedIn](#) • [GitHub](#)

English (C1) • Russian (C1) • Uzbek (Native)

PROFILE

Data Science and ML specialist with hands-on experience in data analysis and preprocessing. Skilled in data cleaning, feature engineering, and building predictive models (supervised learning). I turn raw data into insights, optimize features, and automate analytical pipelines. Actively growing in AI and advanced ML architectures.

EXPERIENCE

AI / ML Trainee

Trainee / AI & ML Engineer

April 2026 — Present

Tashkent, Uzbekistan

- Participated in group ML/AI training sessions covering neural network architectures, activation functions, and optimization methods (gradient descent, backpropagation)
- Built prototype ML models in PyTorch and scikit-learn, evaluating and comparing approaches

Data Analysis & Machine Learning Practicum

Trainee / Data Science Project Developer

February 2025 — March 2026

Tashkent, Uzbekistan

- Developed and optimized machine learning models (including ensemble methods, Gradient Boosting Classifier) using scikit-learn
- Built end-to-end data preprocessing pipelines in pandas to clean, integrate, and structure disparate datasets
- Applied dimensionality reduction and automated feature selection (SelectKBest), speeding up model training without sacrificing accuracy
- Conducted exploratory data analysis (EDA) with matplotlib and seaborn to visualize features and uncover patterns
- Wrote clean, reusable Python scripts to automate data processing, speeding up dataset preparation for modeling

PROJECTS

Agentic RAG Assistant (DeepSeek + BM25) | [GitHub](#) • [Demo](#)

Self-correcting RAG agent built with LangGraph: answers questions about an uploaded PDF in any language, falls back to web search when context is insufficient, and checks its own answer for groundedness before responding.

- Designed a graph pipeline (retrieve → grade_documents → generate → grade_generation) with LLM graders for relevance and groundedness instead of a linear RAG
- Implemented cross-lingual retrieval on top of BM25: automatic document language detection and query translation before search — closing the gap between question and document language
- Added web search (Tavily) as a fallback and a retry limit that guarantees the graph always terminates without looping
- Deployed a Gradio interface with PDF upload and live agent-step display; shipped the service on Render

Stack: Python, LangGraph, DeepSeek API, BM25 (rank_bm25), Tavily, Gradio, Render

Synapse — Multi-Agent AI Analyst | [GitHub](#) • [Demo](#)

General-purpose multi-agent system built with LangGraph: a supervisor routes each question across memory, documents, web search, and a database, while a critic reviews every answer before it's shown.

- Designed a graph of 9 agents (recall, supervisor, retriever, web, data_sql, code, draft_answer, critic, save), where an LLM supervisor with structured output dynamically selects the data source
- Implemented a self-critique loop: a critic agent evaluates the draft's groundedness and completeness and sends it back for revision when needed (with a revision limit)
- Built the backend on FastAPI with streaming event delivery (SSE) — users see each agent's status live
- Built the frontend with Next.js/React/TypeScript and deployed the full stack to production (Vercel + Render)

Stack: Python, LangGraph, LangChain, FastAPI, Next.js, TypeScript, Qdrant, Tavily, Langfuse, Vercel, Render

SKILLS

Programming: Python, SQL

Data Science & ML: Supervised Learning (classification, regression), Unsupervised Learning (clustering), scikit-learn, XGBoost, PyTorch, Deep Learning, Feature Engineering, SelectKBest

Libraries: Pandas, NumPy

Visualization & Analysis: Matplotlib, Seaborn, EDA, Power BI, PostgreSQL, Excel

Tools: Jupyter Notebook, GitHub, VS Code

LLM Platforms & APIs: OpenAI, Anthropic (Claude), Google Gemini, DeepSeek, Prompt Engineering, RAG, Agentic RAG

Agentic AI & Orchestration: LangGraph, LangChain, Multi-Agent Systems, BM25 (rank_bm25), Qdrant, Tavily, Langfuse

Web & Deployment: FastAPI, Next.js, React, TypeScript, Server-Sent Events (SSE), Gradio, Vercel, Render