ENES KUZUCU

Ist | +905064024980 | enesesvetkuzucu@gmail.com | Github | LinkedIn

SKILLS

- **Programming:** Python | C | JAVA | MATLAB | SQL | AMPL
- **Software & Tools:**
  - **LLM:** Langchain, CrewAI, Ollama, LangGraph, LlamaIndex, HuggingFace
  - **ML:** PyTorch, Tensorflow 2, Scikit-learn, BigQuery ML , Prophet
  - **Computer Vision & Image Processing:** OpenCV, StableDiffusion, ComfyUI, torchvision, PyTorch Model Zoo,
  - **Optimization & Parallel:** GLPK, GUROBI, CUDA, Dask
  - **Data Processing & Databases:** Pandas, PySpark, Chroma, Airflow & BigQuerry, MongoDB, PostgreSQL, MySQL
  - **Other:** Git, Flask, FastAPI, Django, Docker, Kubernetes, AWS, GCP
- **Languages:** English (TOEFL IBT 90) | Turkish | Russian

- PyPI packages: LLMService proteas facexformer-pipeline categorizer serpengine brightdata

EXPERIENCE

Data Scientist *EETech, USA (Remote)* Feb 2021 – Aug 2025

- **LLM Pipeline for Data Imputation and Normalization**: Developed custom data imputation pipeline. The core goal was to leverage SERP and Web Scraping and LLMs to find missing information about products. I already had open source packages related to scraping and LLM based data extraction. We combined 3 modules for our custom use case and serve the pipeline over fastAPI. I also added context aware mode where LLM uses injected context for data extraction. This application solved a chronic problem for electronic manufacturers and distributors and is currently used by two distributors.
  Since concurrency limitations of each submodule were different for different reasons, I used a custom asycn logic to allow faster operations. .
- **LLM-based Document Processor - RAG - Application**: Created a python workflow for processing datasheets (technical PDFs) aimed at reducing work hours spent by contractors on manual information extraction. I was responsible for the initial Literature Review and coming up with fast solutions. We explored different LLMs and tested their data extraction capabilities using custom tailored metrics with labeled mini datasets. These tests included VLLMs and local modals.
  Used Chroma as a vector database and worked on different chunking strategies. In the end we were able to reach around 55% collective similarity score for existing data and 9% collective similarity score for tests where we test data extraction with non-existing data irrelevant.
  We started using a context module for each page in pdfs. Some context from the very first page and last page is injected and this helped greatly increase our collective similarity score to %68 percent.
  Although there are no official measures, our metrics reach a 80% reduction in hours for specific tasks like Product Change Notice (PCN) extraction.
- **Demand Forecasting and Custom Data Quality Metrics for Time-Signals**: Developed a multi-model demand forecasting application, achieving 93% accuracy for increase/decrease predictions based on 3 months of cross-validation results. Created custom data quality metrics, such as data integrity score, time series cohesion score, and predictability score, to segment datasets based on their forecastability.
- **Data Cleaning Strategy for Electronic Component Dataset**: Implemented a robust data cleaning strategy for a crawled electronic component dataset consisting of 120 million rows. Developed multiple custom tailored filters and created a pipeline mechanism using the Prefect package to process them. Successfully eliminating 98% of noise while retaining 96% data coverage. Used Pandas and Spark extensively
- **Team Growth in Turkey**: Responsible for expanding our developer team in Turkey, I collaborated with three recruiter agencies and conducted interviews for various positions, including technical data science interviews. Successfully grew the team size from 2 to 9 members.

Founder / Lead Backend Engineer budgety.ai July 2024 - May 2025

- At its core, budgety.ai is a tool which extracts and categorizes records from bank statement PDFs for personal finance management via an LLM-driven pipeline.
- Split the backend into four parts:
  - Table extraction from documents: This was challenging due to the lack of standardization across bank statement files and variations between banks. Not all PDFs work with pdfminer under the same configuration. I devised a complex workflow to process PDFs through various libraries, including OCR, as well as AWS services..
  - Table normalization: Normalizing table column names, and date and amount/spending related columns. It is harder than it sounds because you have to tackle with various of conventions regarding date and amount

    - Categorization**:** Developed a custom LLM-based categorizer. Categories are defined in a YAML file, and the system categorizes input strings accordingly. I then extended it to handle batch processing with threading, later converting it to asyncio-based batch support, and finally added keyword-based and pattern-based fast-path solutions.
    - FastAPI serving**:** Built the entire backend from scratch using FastAPI and SQLAlchemy, with a modular service layer, asynchronous background workers, and JWT authentication. Deployed a fully containerized CI/CD pipeline on AWS ECS, employing repository and service patterns, background tasks, and dependency injection for cleaner architecture.
- The biggest challenge was productionizing the app. It was humbling in terms of understanding and managing the entire business aspect

Backend Developer & Data Scientist *TeamProcure Enigma Prototype* August 2024 - October 2024

- Led a team to deliver a working prototype of Enigma, an **LLM-powered chatbot** that **converts natural language into SQL**, runs queries against a 100-table test database, and **generates visualisations of the results.**
- I created schema documentation and implemented a RAG pipeline to retrieve relevant information. I used a different configuration to optimize RAG. It worked better but results were not feasible.
- End solution: I restructured the schema documentation in a more indexable way and I used mini LLMs as agents for marking relevant parts via their indexes. And then Used marked indexes to recompile only the relevant parts. This resulted in x2 elapsed time but also the overall cost diminished around 1/16.
- Also the end result was able to create accurate visualisations with accurate diagram types.

EDUCATION & PUBLICATIONS

Master's of Science in Big Data & AI 08/2020 - 07/2022
Novosibirsk State University, Russia

Bachelor of Biomedical Engineering 09/2014 - 06/2018
Kocaeli University, Turkey

"Button Press Detection from EEG Signals Using Deep Learning", 2022 IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM), https://doi.org/10.1109/EDM55285.2022.985519