Sejal Barshikar

barshikar.s@northeastern.edu | LinkedIn | GitHub | Portfolio

EDUCATION

Northeastern University **Aug 2025 - Dec 2027**
MS in Computer Science GPA: 3.83/4

Savitribai Phule Pune University **Aug 2021 - May 2025**
BE in Artificial Intelligence and Data Science GPA: 3.46/4

SKILLS

Programming Languages : Python(NumPy, Pandas, scikit-learn, PyTorch, TensorFlow), Javascript, SQL, C++
AI & Machine Learning : CNN, Transformers, Supervised Learning, Unsupervised Learning, RNN, NLP, A/B Testing, RAG, Random Forest
Database Technologies : MySQL, PostgreSQL, Oracle, RDBMS, NoSQL
Big Data & Cloud : AWS, Azure, GCP, Databricks, PySpark, Hadoop, Snowflake
Frameworks & Models : LangChain, BERT, BART-CNN, LLaMA, OpenAI Whisper, CLIP, YOLO, OpenCV
Tools : Git, FastAPI, Flask, REST APIs, CUDA, MLflow, NLTK, Docker

PROFESSIONAL EXPERIENCE

AICTE (All India Council for Technical Education) | Data Science Intern **Dec 2024 – May 2025**

- Built an **end-to-end ETL** pipeline processing **5M+** retail transaction records using optimized **Pandas** workflows, reducing data preprocessing time by **60%**
- Engineered **RFM behavioral features** (recency, frequency, monetary) and applied **K-Means** clustering evaluated across **10+** initializations via **Silhouette** analysis, identifying **5** distinct customer segments
- Collaborated with a team of **10+** analysts and engineers to define segmentation criteria, validate **cluster profiles**, and align model outputs with **targeted marketing** objectives
- Deployed segmentation inference via **Flask REST API** achieving **sub-100ms** latency for real-time usage
- Drove **20%** increase in conversion rates through segment specific behavioral profiling and **personalized** recommendations

SPPU (Savitribai Phule Pune University) | Research Under Prof. Swati Kadu **Jan 2024 – Sept 2024**

- Designed a **MAML** based meta-learning framework for retrieval-augmented **code summarization** across **108K** Python code-summary pairs, outperforming CodeBERT by **16%** on BLEU-4
- Engineered a **two-stage** coarse-to-fine retrieval pipeline combining **TF-IDF** filtering with **BERT semantic re-ranking**, improving METEOR by **34%** and ROUGE-L to 43.06 over pretrained baselines
- Collaborated with a team of **5** researchers to benchmark **4** competitive baselines, evaluating generalization across procedural, OOP, and functional Python paradigms

PROJECTS

Video Content Analyzer | Demo

- Engineered a **multimodal** video analysis pipeline integrating **YOLOv8 + BoTSORT**, **Whisper**, and **CLIP** to automate semantic extraction across vision, speech, and scene modalities simultaneously
- Achieved **6x** faster than real-time CPU inference, processing a 6-minute video in **62 seconds** with peak memory under **1.7GB** enabling deployment on memory-constrained environments
- Architected a **FastAPI** + **Docker** backend with **async** processing, concurrent video uploads and sub-500ms API response times

VentSpace | GitHub

- Engineered **audio emotion** pipeline training **SVM** on 35 **librosa** features across 1,056 labeled samples from RAVDESS and CREMA-D achieving **84%** accuracy with **0.87** precision on frustration class
- Integrated **OpenAI Whisper** transcription with cycle-aware context to deliver **personalized** weekly mental health reports via a **multimodal** fusion pipeline
- Designed a public **REST API** with structured **JSON** output across **5** fields enabling clean multimodal fusion across **audio, text, and cycle-aware pipelines**

Research Paper Classifier (Fine tuning BERT) | Github

- Analyzed **28K** research papers to classify **11** categories with **80.3% accuracy**, identifying semantic overlaps to inform data quality improvements
- Conducted systematic **error analysis** via confusion matrix identifying **semantic overlap** between interdisciplinary categories, documenting category-specific failure modes to guide future **dataset refinement**
- Optimized training for Apple Silicon MPS with **gradient clipping**, linear warmup scheduling, and **AdamW**, achieving stable convergence in a single epoch across **3,549** batches on a **resource-constrained** environment