Seoul, Korea
siksicking@gmail.com

Sunsik Kim

Data Scientist

linkedin.com/in/sunsik-kim

Data Scientist with 5 years at Bunjang (Korea's leading C2C marketplace, 10M+ MAU), building production ML systems on AWS, and recently focused on stateless, high-throughput LLM inference optimized for cost and latency at scale. I'm now looking to build richer, more stateful LLM applications — agentic systems in particular — bringing the same rigor around evaluation and production reliability to a wider range of problems.

Work Experiences

Data Scientist Bungaejangter Inc. (dba Bunjang) 2021 — Present

Projects

In-Chat Abuse Detection System **December 2025 — July 2026**
Bunjang *Seoul, Korea*

- **Result**: Delivered a **4.6x increase in abuse detection coverage** over the legacy regex/OCR system, and **49.5% more sanctioned users** after full rollout with false-positive rate held within threshold — driving sanctioned users' subsequent transactions to escrow and capturing GMV previously occurring off-platform.
- **System**: Implemented the batch producer and LLM worker in an **event-driven inference architecture** on AWS, where the batch producer queries chat logs via **Athena** and publishes detected events to SQS for consumption by LLM workers running on an EKS cluster, built for a future batch-to-real-time migration.
- **Model**: Designed a **4-stage cascaded LLM inference pipeline**, decomposing a monolithic instruction into task-specific stages to lift precision on production-representative test sets; selected **gemma-4-26B-A4B-it** after benchmarking 5 open-source candidates, replacing the external LLM API with a managed-hosting endpoint to bring operating costs within budget.
- **Infrastructure**: Benchmarked a **self-hosted vLLM deployment** on an AWS **DLAMI g7e.2xlarge** GPU instance against the incumbent managed LLM hosting service — same pipeline, both setups — showing a **60% wall-time reduction** at precision parity within margin of error, the quantitative basis for migration planning.

AI-Assisted Product Registration System **January 2025 — June 2025**
Bunjang *Seoul, Korea*

- **Result**: Versus a non-AI control cohort, drove a **4pp increase in weekly registration conversion** and cut time-to-first-listing of new users by 13.5%. Among AI-assisted users, **listings per user rose 47.2%** post-launch, and 85% of survey respondents called the feature helpful.
- **System**: Shipped a product registration system with **five field-specific components** (title, category, options, description, price), each orchestrating its own model and downstream service by input/output modality (image, text, multimodal).
- **Model**: Used an LLM for open-ended language tasks (title, description, options), and fine-tuned a **lightweight in-house classifier** for category from an existing Korean text encoder, tuned to match the four-slot UI carousel (HitRatio@4) — reserving the LLM for tasks that actually needed it.
- **Pipeline**: For price — the one field needing signals from comparable listings, not just item attributes — built a **multi-stage pipeline** chaining an LLM (query generation), an in-house search engine, and a reranking-and-statistics service to derive a range from live comparables.

Similar Product Recommendation System **March 2024 — October 2025**
Bunjang *Seoul, Korea*

- **Result**: Drove **10.2% CTR / 13% PDP-entry gains** on the home recommendation surface, and **4.1% CTR / 9% PDP-entry gains** on the product-detail surface after full rollout; PDP-entry user share rose **10.4%** and **3.6%** respectively.
- **System**: Designed and implemented a **hybrid product embedding** unifying title (text encoder), category-brand (collaborative filtering on user-item interactions), and price/size (sinusoidal positional encoding) into a single vector for real-time similarity search over a multi-million-item catalog.
- **Model**: Pretrained a lightweight Korean text encoder from scratch on Bunjang product text with **ELECTRA** via **PyTorch**/**Hugging Face** transformers, then **supervised SimCSE** domain alignment on (search query, product title) pairs — achieving retrieval parity with a 20x larger public-model baseline at **4–5x lower CPU cost per request** on AWS **Graviton** (c7g.large).
- **Infrastructure**: Implemented the **hourly indexing batch** job that populates **Aurora** pgvector (the team's choice over Milvus and **S3 Vectors**), triggered as a **SageMaker** Training Job via MWAA-orchestrated Airflow. Built the **embedding API** as a containerized FastAPI service, pushed to ECR and deployed on EKS.

Education

Master of Statistics, *Korea University* 2019 — 2021
Bachelor of Economics (Statistics), *Sungkyunkwan University* 2013 — 2019