Mingyu Kim

Education

2017.03 – 2023.09	B.S. in Industrial Engineering, Minor in Statistics, Pusan National University
2023.09 – 2024.03	AI Engineer Training Bootcamp, Modu Research Institute

Work Experience

2024.04.01 – Present	Visuworks Co. | Data Scientist

Summary

- Led 3 AI/ML projects, gaining end-to-end experience from development to deployment and operation.
- Skilled in exploratory data analysis, feature engineering, and addressing limitations of existing approaches by defining problems and developing practical solutions.
- Experienced in building robust data pipelines, deploying ML services in production, and monitoring performance to ensure reliability.

Projects

Lenze size recommendation 2025.03 – Present (6 months)

- Achievement

Quantified prediction uncertainty using causal inference to evaluate prediction reliability for unobserved lens sizes and ensured model stability through integrated data pipeline and monitoring system

- Role/Solution Process

– Prediction Evaluation Problem Solving: Resolved limitations in evaluating predictions for unobserved treatments by leveraging causal inference's positivity assumption and propensity scores, expressing prediction uncertainty through intervals using CQR and partial identification

– Data Consistency Assurance: Addressed data inconsistency between training and inference environments by building integrated data pipeline and feature store, standardizing data preprocessing processes and ensuring consistency across environments

– Monitoring System Development: Developed automated monitoring pipeline using Airflow for real-time data drift detection and performance tracking, enabling early detection of silent failures and ensuring service stability

– Domain-Specific Modeling: Enhanced model interpretability through clinically meaningful feature engineering and monotonicity constraints aligned with physician intuition, and provided model explanations using Shapley values to increase medical staff trust

Chatbot 2024.11 – 2025.03 (4 months)

- Achievement

Automated customer service operations through RAG pipeline implementation (classification model integration, search accuracy improvement) and monitoring system development

- Role/Solution Process

– Led end-to-end process from problem definition to RAG pipeline design and implementation for automating repetitive customer service tasks

– Improved user satisfaction by implementing classification model at pipeline frontend and developing context-aware question rephrasing and keyword extraction features, reducing unsatisfied feedback by over 50%

– Built Qdrant-based VectorDB system enabling users to select document versions and perform real-time updates

– Established monitoring system using RAGAS framework integrated with Airflow for silent failure detection and performance metric tracking to minimize maintenance costs

OCR Pipeline 2024.07 – 2024.10 (4 months)

- Achievement

Achieved error rate below 1% and 99%+ OCR accuracy by implementing object-oriented design and unit testing framework

- Role/Solution Process

– Refactored procedural code into object-oriented architecture to clearly separate functional responsibilities and improve debugging efficiency

– Built unit testing and type validation system using pytest and mypy to prevent runtime errors proactively

– Improved OCR accuracy from 70% to 99%+ by applying domain-specific models tailored to fixed regions per device type and optimizing image preprocessing/postprocessing logic

– Established clinical validity verification and anomaly detection monitoring system to prevent silent failures and ensure stable service operations

Details

Lenze size recommendation

2025.03 – Present (6 months)

- Project Overview:

Developed a service that predicts surgical outcomes for vision correction, enabling surgeons to select optimal lens sizes based on data-driven predictions rather than solely on experience and intuition.

- Problem-Solving Process:

1. Model Validation Challenge

- Problem
Faced challenges in evaluating prediction accuracy when the model generated four size options but only one could be validated with real customer data.

- Solution
Leveraged causal inference (positivity assumption) to exploit distributional overlap across treatments (lens sizes), facilitating indirect performance evaluation. Accounted for non-random treatment assignment (lens sizes) by quantifying overlap in the data and separating overlapping from non-overlapping regions for analysis. Validated this approach not only through domain expertise but also by statistically examining the distribution across treatments.

Used prediction intervals where treatment distributions overlapped and applied partial identification to quantify prediction ranges in non-overlapping regions. Motivated by the goal of increasing user trust, expressed predictive uncertainty explicitly through intervals.

- Result
Indirectly evaluated unobserved predictions and visualized both guaranteed and non-guaranteed ranges to effectively communicate predictive uncertainty.

2. Data quality and consistency challenges between training and inference environments

- Problem
In the absence of a unified data pipeline, training and inference each pulled data directly from the data lake, leading to inconsistencies in data processing between environments.

- Solution
Developed a robust data pipeline to cleanse both OCR-collected and external data sources, incorporating data validation and schema enforcement to ensure data quality.
In addition, built a feature store that unified offline (training) and online (serving) features, securing data consistency across environments and improving the reliability of model deployment.

- Result
Secured data quality through the new pipeline and established consistency across training and inference via the feature store, increasing confidence in model outputs and stability in deployment.

3. Modeling Approach

- Problem

The goal of model development was to deliver results that achieve user satisfaction, thereby fostering trust.
This required not only improving predictive accuracy within the dataset, but also ensuring robust performance in inference settings.
In addition, the model needed to produce outputs aligned with user intuition—particularly important as the primary users were physicians, necessitating medically interpretable predictions.

- Solution

To improve model performance, I placed strong emphasis on thoroughly examining the data.
I first analyzed feature distributions and filtered out implausible values that deviated from clinical standards, ensuring a reliable dataset.

Drawing on physician intuition, I engineered clinically meaningful features—for example, capturing the relationship between age and lens thickness—which not only improved predictive accuracy but also aligned the model's behavior with medical reasoning.

To further strengthen reliability, I incorporated medical domain knowledge into the model by applying monotonicity constraints where clinically appropriate.

Finally, I used interpretability tools such as Shapley values to explain prediction outcomes, enabling physicians to better understand and trust the model's decisions.

- Result

Through detailed data analysis, I improved data quality and developed new features that enhanced model performance while producing results consistent with medical intuition.
In addition, the model was not treated as a black box; I provided explanations for its outputs, ensuring that the reasoning behind predictions was transparent and understandable.

4. Lack of monitoring for silent failures and model performance

- Problem

Monitoring was required to ensure model performance in inference settings.
It was important to track not only how well the model performed during real-world usage, but also to detect potential issues such as data drift.
Without such monitoring, the service could appear to function normally while causing user discomfort through silent failures, ultimately reducing user trust and engagement.

- Solution
Developed an automated monitoring pipeline with Airflow to detect data drift using statistical measures (e.g., Jensen-Shannon distance) and to continuously compare training vs. inference data distributions.
Implemented performance tracking by monitoring prediction-outcome gaps in real time, enabling early detection of silent failures and ensuring reliable model performance in production.

- Result
Enhanced service stability and maintained user trust by proactively identifying data drift and performance degradation before they impacted end users.

Chatbot

2024.11 - 2025.03 (4 months)

- ### Project Overview

Developed an AI chatbot service to automate repetitive customer inquiries, reducing the burden on human agents and improving overall customer service efficiency. The system processed an average of 100-200 daily inquiries, significantly enhancing agent productivity.

- ### Problem-Solving Process

1. Improving User Satisfaction through Classification

- Problem
User feedback revealed that the most common source of dissatisfaction was the LLM generating ambiguous responses to questions it couldn't clearly answer. This was particularly problematic when customers asked about services that required manual intervention (e.g., appointment scheduling with personal information), yet the system continued to engage in unnecessary back-and-forth conversations.

- Solution
Implemented a classification model at the front-end of the RAG pipeline to categorize incoming questions and route them appropriately. Used LLM-based classification prompts to identify question types and ensure proper handling of different inquiry categories.

- Result
Reduced "unsatisfied" feedback by over 50% by preventing inappropriate responses and unnecessary conversations.

2. Context-Aware Document Retrieval

- Problem
The system needed to maintain conversation context to provide relevant responses. For example, when a user asked "What's the price of LASIK?" followed by "What about LASEK?", the system should understand the second question refers to LASEK pricing.

- Solution
Implemented question rephrasing using Redis to store short-term conversation history. The system retrieves conversation context and rephrases questions to match the conversation flow, enabling more accurate document retrieval.

- Result
Improved response relevance by maintaining conversation context and reducing the need for users to repeat information.

3. Enhanced Search Performance through Query Decomposition and Keyword Extraction

- Problem
Single questions often contained multiple semantic components (e.g., "Tell me the prices of both LASIK and LASEK"), requiring retrieval of documents for multiple sub-questions. Additionally, semantic search limitations and embedding model constraints made it difficult to find documents containing proper nouns like doctor names.

- Solution
Implemented query decomposition to break complex questions into multiple sub-queries, each targeting specific information needs. Added keyword extraction to complement semantic search, enabling filtering based on extracted entities like proper nouns.

- Result
Improved search accuracy for complex queries and enhanced retrieval of documents containing specific entities that embedding models might miss.

4. Document Retrieval Optimization

- Problem
The core challenge was creating high-quality embedding vectors for semantic search to retrieve the most relevant FAQ documents.

- Solution
Built a test dataset from FAQ data using LLM-generated questions to evaluate different embedding models. Selected the best-performing model through systematic comparison. Implemented Qdrant as the vector database with real-time updates when documents are modified, minimizing maintenance costs. Used multi-tenancy features to clearly separate document versions.

- Result
Achieved optimal document retrieval performance with a maintainable and scalable vector database solution.

5. Response Generation

- Problem
Generated responses needed to be both accurate (based only on retrieved documents) and user-friendly, maintaining a conversational tone that customers would find approachable and easy to understand.

- Solution
Applied prompt engineering to constrain the LLM to generate responses based solely on retrieved document content while maintaining a friendly and accessible tone. This dual focus ensured both accuracy and positive user experience.

- Result
Delivered responses that were both factually accurate and user-friendly, improving overall customer satisfaction.

6. Monitoring and Maintenance

- Problem
Silent failures in the chatbot system could go undetected, potentially causing user discomfort and reducing trust in the service. Continuous monitoring was needed to ensure reliable performance and minimize maintenance costs.

- Solution
Implemented an automated monitoring system using the RAGAS framework to evaluate response quality. The system scores responses based on their grounding in retrieved documents and tracks performance metrics. Used Airflow to process chat logs from spreadsheets, generating daily and weekly performance reports and alerting high-scoring conversations via Slack.

- Result
Proactively detected silent failures and performance issues, ensuring reliable service operation and maintaining user trust through continuous quality monitoring.

OCR Pipeline 2024.07 – 2024.10 (4 months)

- ### Project Overview

Developed a robust OCR pipeline to extract test results from medical images and store them in a structured database.
Since the pipeline served as a critical component for collecting data used in production services, it was designed with a strong focus on real-time processing, stability, and high accuracy.

- ### Problem-Solving Process

1. Ensuring Stable Service Operations

- Problem
Inherited a legacy project with a large backlog of error logs. The codebase was written in a purely procedural manner, making it extremely difficult to trace the root causes of errors. The lack of clear structure and responsibility boundaries hindered both debugging efficiency and long-term maintainability.

- Solution
Refactored the system into an object-oriented architecture, applying core principles such as single-responsibility per function and well-defined ownership for each object. This restructuring enabled clearer separation of concerns and improved readability.

Additionally, integrated unit testing with pytest and static type checking with mypy to proactively detect potential runtime errors. Together, these practices established a robust foundation for reliable and maintainable service operations.

- Result
Achieved an error rate below 1%, while significantly improving debuggability.
Even when issues occurred, the modular design and testing framework enabled rapid root-cause identification and resolution, ensuring stable service delivery and reducing operational overhead.

2. Improving OCR Accuracy

- Problem
The existing deployed OCR model had been developed without a proper test set and was only validated manually by its original developer.
To rigorously evaluate performance, I constructed a dedicated test dataset covering ~100 samples for each of the six different diagnostic devices.

Benchmarking revealed that the deployed model achieved less than 70% accuracy, which was far below the reliability required for production use.

In addition, silent failures in production (e.g., plausible but clinically invalid outputs) were not being detected, creating risks for downstream systems and users.

- Solution
Leveraged domain-specific characteristics to boost performance. Although the image types varied across devices, the regions of interest to extract were fixed per device type.
This allowed the use of rule-based localization to reliably identify the relevant regions, effectively simplifying the text detection step.

For text recognition, integrated TrOCR, an open-source model specialized in OCR tasks. Model evaluation with the curated dataset showed accuracy around 95%, which was further improved to over 99% by applying tailored image preprocessing and position-specific postprocessing strategies.

To address silent failures, designed a monitoring layer that applied clinical plausibility checks and probabilistic anomaly detection. The system raised alerts whenever extracted values were outside valid ranges or statistically improbable given the device type, enabling proactive detection of hidden errors.

- Result

Achieved over 99% OCR accuracy in a mission-critical pipeline while also ensuring reliability through monitoring. The monitoring layer consistently surfaced out-of-range or anomalous values, reducing undetected OCR errors and enabling faster triage when issues occurred.

These improvements ensured stable service operations, minimized error propagation to downstream systems, and reinforced trust in the automated data pipeline.