Minseok Oh,

Data Scientist Portfolio

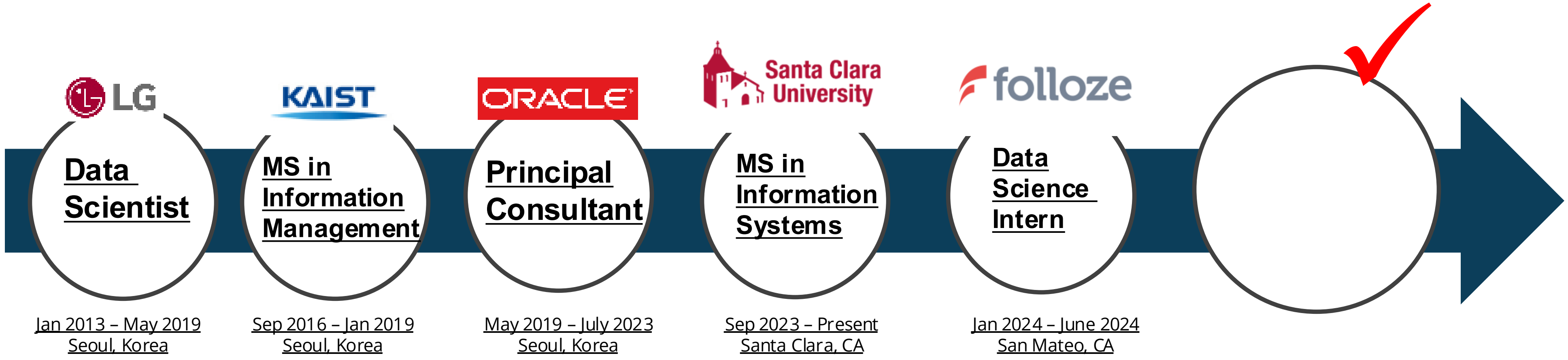
MSIS at SCU (Expected June 2025)
Ex-ORACLE, LG

Data Scientist with 10+ years of experience specializing in ML/DL solutions that drive business value. Track record of delivering hig h-impact projects through experimentation and data-driven deci sion making.

Cupertino CA, 95014
(408) 334 – 5898
ohmseok0524@gmail.com
https://www.linkedin.com/in/scottmsoh/

Career Journey

✓ **Driven to push boundaries in data science across industries, from manufacturing to enterprise solutions**
Now ready to create transformative impact through innovative data-driven solutions in Silicon Valley



✓ Started career as a **Data Scientist at LG**, working on **process optimization and sentiment analysis using NLP** in the manufacturing sector. To broaden my experience beyond manufacturing, I pursued an **MS in Information Management at KAIST**, where I worked on **machine learning and data analytics projects** across **education, e-commerce, and healthcare**, gaining hands-on experience in **A/B testing and experimental design**.

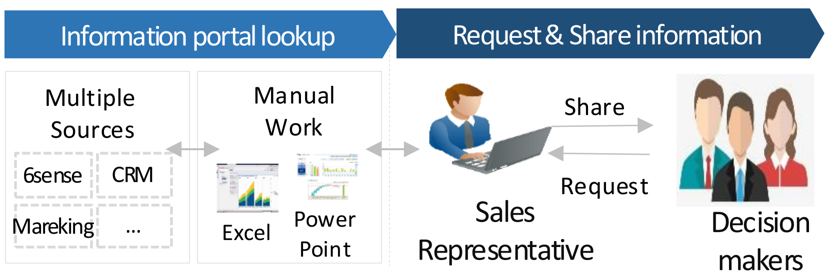
✓ After graduation, I joined **Oracle**, expanding my expertise with **enterprise-scale projects** in **finance, insurance, retail, and pharmaceuticals**, focusing on the predictive modeling such as **recommendation systems, demand forecasting, and churn prediction**. Seeking **further growth and diverse challenges**, I moved to **Silicon Valley** to advance my career in data science.

CONTENTS

✓ **Transitioning from manual, experience-based sales decision-making to a data-driven approach using machine learning techniques for improved accuracy and efficiency.**

As-Is

Current challenges among sales process



Qualitative forecasting of demand just based on past experience

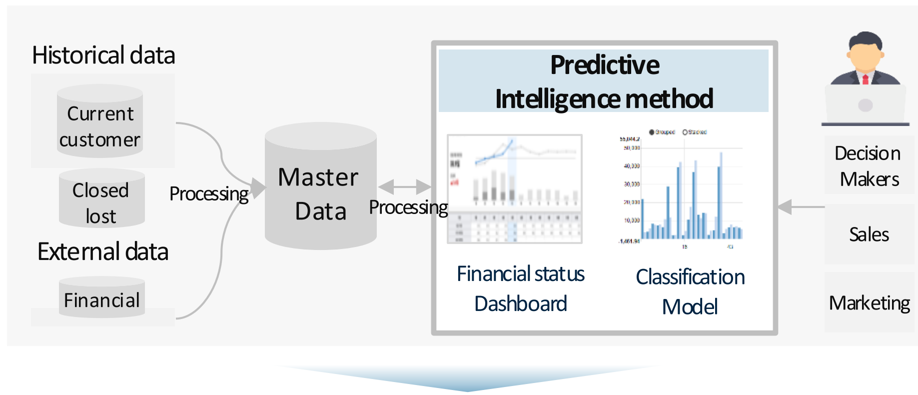
Non-consideration of external factors such as financial fluctuations

A significant number of sales employees participate in the planning process

Lack of agility in providing data for reporting and decision making

To-Be

Data-driven decision-making Process



- Support accurate data-based decision-making through change s in financial position by company and past data inquiry
- Enable accurate decision-making through accurate future cust omer classification of machine learning classification models

	Customer Propensity Analysis for Sales Forecasting and Targeting	
1	**Business pain-point**	Limited sales resources make it difficult to manage all customer lists. ➢ **Aim to focus on high-potential customers by predicting potential vs. non-potential customers.**
2	**Project Overview**	This project aims to develop a machine learning classification model to identify potential future customers. ➢ **Optimizes lead prioritization, enhances customer targeting, improves sales efficiency, and drives business growth.**
3	**Data Collection**	• Customer & Company Data: ➢ **Demographic/transactional information of clients and non-clients.** • External Financial Data: ➢ **MRDS dataset containing company profiles and financial indicators.** • Feature Set: ➢ **130+ features, including company metadata (e.g. industry, contract details) and financial metrics (e.g., revenue, credit rating).**
4	**Data Preparation**	• Imbalanced Data: ➢ **Class imbalance between customers (less than 10%) and non-customers.** • Missing Value Imputation: Applied median, mean, and mode where necessary ➢ **Tree-based model used for imputing crucial missing values (SPC SRC: D~A+)** • Scaling & Transformation: ➢ **Used Robust Scaler and Log Transformation for skewed data.** • Outlier Detection & Treatment: ➢ **Identified and handled outliers using IQR**
4	**Exploratory Data Analysis**	• Examined key financial metrics such as revenue, liabilities, and stockholder equity. • Identified correlations between features impacting customer classification ➢ **Remove multicollinearity using VIF, PCA, Regularization(L1) techniques** • Visualized distribution trends across different customer segments ➢ **Small, medium, and large customers, each capturing unique characteristics specific to their segment.** ➢ **Ensemble models (Develop 3 different models)**

	Customer Propensity Analysis for Sales Forecasting and Targeting	
5	**Feature Engineering**	• Create new features: lifetime value, Report filing delay, Gap preliminary official
6	**Manage Imbalance data**	• Consider and experiment with different methods to address data class imbalance: **1) SMOTE (Synthetic Minority Over-sampling Technique)** 2) Random oversample 3) Random undersample 4) Class weight (Cost-sensitive modeling)
7	**Model development & evaluation**	• Experimented with and trained various models, including SVM, Decision Tree, Random Forest, XGBoost, LightGBM, AdaBoost with Decision Tree, Logistic Regression, and an Ensemble Voting model ➢ **3 XGBoost models with SMOTE (for small, medium, and large customers) ensembled using weighted majority voting (Tried stacking as well, but it led to overfitting and was too complex to maintain)** ➢ **Recall is crucial to avoid missing potential customers, ensuring maximum lead capture. A recall of 0.82 minimizes lost opportunities, while an F-beta score of 0.83 balances precision and recall** • To Reduce Overfitting: **PCA, Regularization (L1, L2), VIF (Remove features bigger than 8.0)**
8	**Hyper parameter tuning**	• Fine-tuned hyperparameters to mitigate overfitting and enhance performances ➢ **max_depth = 5** ➢ **Eta = 0.2** ➢ **Colsample_bytree = 0.9** ➢ **Colsample_bylevel = 0.7**
9	**Reflections and Future Improvements**	• There were many private clients, but the lack of public financial data required exploring alternative solutions • Aimed to apply the model to the business through A/B testing, but due to time constraints, we handed it over to the responsible team

- ✓ **Designed a 3-stage re-ranking architecture incorporating SGD-based matrix factorization for collaborative filtering, optimizing item recommendations based on store-specific sales volume**

✓ SGD matrix factorization

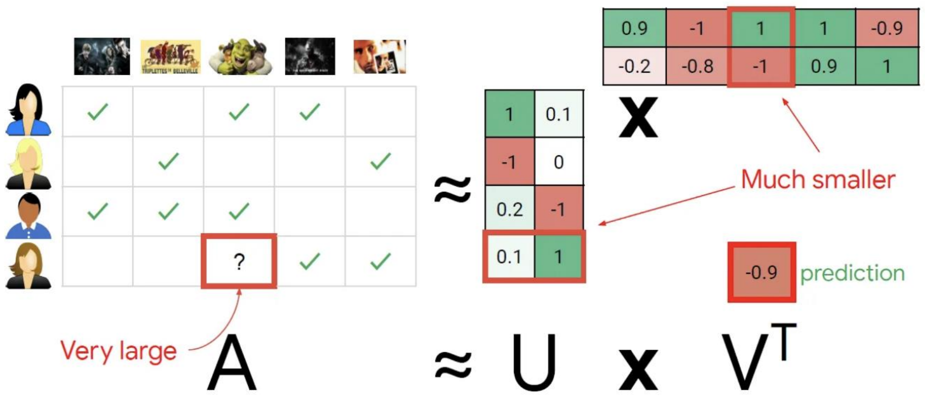
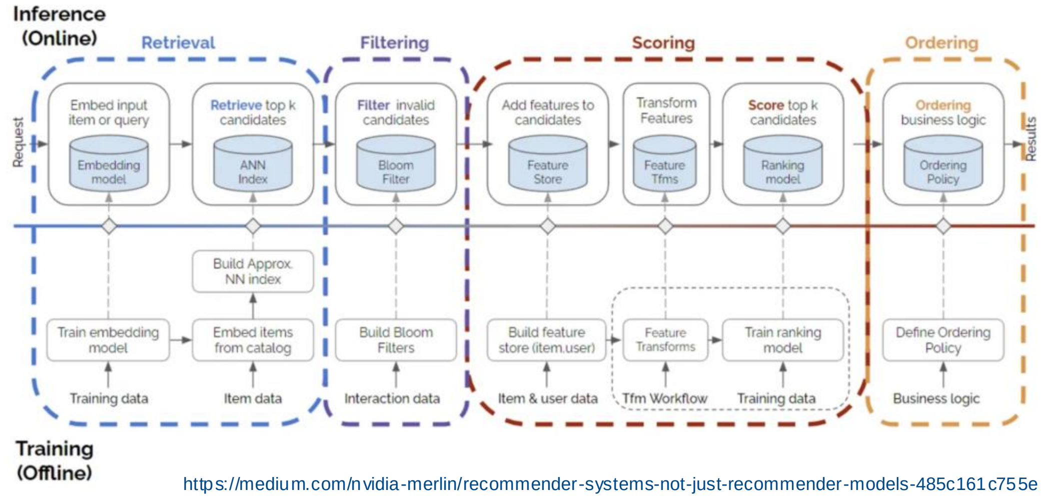


Image: Google Cloud

- Factorizes the store-item interaction matrix into lower-dimensional latent factors.
- SGD (Stochastic Gradient Descent) is used to optimize the factorization.
- Predicts missing interactions (e.g., ratings, purchases) based on learned embeddings.

✓ 3-stage recommendation system architecture



https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e

- Retrieval: Fetches top-N relevant candidates using an ANN index.
- Filtering: Applies Bloom Filters to remove invalid recommendations.
- Scoring & Ordering:
- Enhances rankings with additional feature transformations.
- Uses a ranking model (e.g., ML-based scoring).

	Optimizing Store-Specific Item Recommendations with a 3-Stage Re-Ranking Architecture	
1	**Business pain-point**	Faced stagnant sales growth and sought new revenue opportunities by leveraging machine learning and a data-driven recommendation system to o ptimize inventory management and enhance product sales.
2	**Project Overview**	• Tried to propose market basket analysis using POS sales data • Ultimately designed a 3-stage re-ranking architecture with SGD matrix factorization-based Collaborative Filtering (CF) recommendation model to suggest items based on store-specific sales volume.
3	**Architecture**	Designed a 3-stage Re-Ranking Architecture ➢ Stage 1: SGD-based Matrix Factorization – Generates initial item recommendations. ➢ FAISS Indexer: Enhances efficiency by enabling fast similarity search. ➢ Stage 2: Bloom Filter: Filters out items already sold at the store. ➢ Stage 3: XGBoost Ranker – Refines rankings based on additional features.
4	**Data Collection**	• 27 data tables (Supplier, customer information, POS data and so on) ➢ **Utilized store-specific item sales quantity for recommendations.** **(Since they operate only offline stores, they lack customer interaction data)**
5	**Data Preparation & Feature engineering**	• Sparse Matrix Construction – Converting raw transactional data into a store-item interaction matrix. • Feature Engineering for Stage 2 Re-Ranking Model : Used around 40 features ➢ Store Average Sales Amount The mean sales given by a store across all items on sale. ➢ Item Average Sales Amount The mean sales an item received from all stores. ➢ Store-Store Cosine Similarity Measures how similar a store is to other stores based on sales patterns. ➢ Item-Item Cosine Similarity Measures item similarity based on shared store interactions. ➢ Rating Count Features Number of sales per store and sales per item, which help assess data density and confidence levels.

	Optimizing Store-Specific Item Recommendations with a 3-Stage Re-Ranking Architecture	
5	**Model selection & development**	Consider of four recommendation models: • Apriori with Association Rules → Excluded ➢ Difficult to set optimal lift and support thresholds, automation challenges. • FP-Growth → Excluded ➢ Interpretation of tree-based patterns is challenging • **Collaborative Filtering → Adopted (Benchmarked from research papers on store-item sales-based recommendation models)** ➢ KNN-based CF: Too simple; performance was suboptimal ➢ **SGD-based Matrix Factorization: Suitable for capturing high-dimensional interaction patterns.** • Deep Learning-Based Models → Excluded ➢ Lack of organizational expertise in machine learning/deep learning ➢ No available infrastructure to support deep learning deployment.
6	**Model Evaluation**	Optimizes score calibration for fine-tuning the ranking, where metrics NDCG: NDCG@10: 0.79 → Top 10 item selection
7	**A/B test & Result**	• Base Metric: Daily Revenue • Expected Metric: 5% Increase in Daily Revenue • Controlled Conditions ➢ Same Store Locations ➢ Same Day of the Week (to control demand fluctuations) ➢ Same Time Window (to eliminate hourly sales variations) • Statistical Parameters: ➢ Power (1-β): 85% ➢ Significance Level (α): 5% ➢ Experiment period (Based on minimum Sample Size): 31.36 days (calculated using t-test formula). • A/B Test Results: ➢ Comparison by Time Zone & Item Category ➢ Lift & p-value analysis to determine statistical significance. ➢ **Revenue (Monitoring Metric) shows a Significant Positive Impact (+4.5%)** ➢ Indicates Quantifiable Growth due to product recommendations ➢ Confirms Positive Business Impact from optimization in recommendation strategy

Outline Early Warning System for Customer Churn

Define Scope

- Identify potential churning customers early
- Develop preventive measures for retention
- Create actionable insights for sales team
- Establish monitoring system for at-risk customers

Machine Learning Process

Input Data

- Customer data
- Financial data
- Behavioral data
- Policy information

Data Engineering

- Handle imbalanced data
- Missing value treatment
- Outlier detection
- Feature engineering

Modeling

- Classification implementation
- Hyper parameter tuning
- Performance Metrics
- Overfitting prevention

Output

Provide insights on predicted user churn

Due to infrequent updates in customer data, predictions are reported weekly, with model retraining scheduled every month

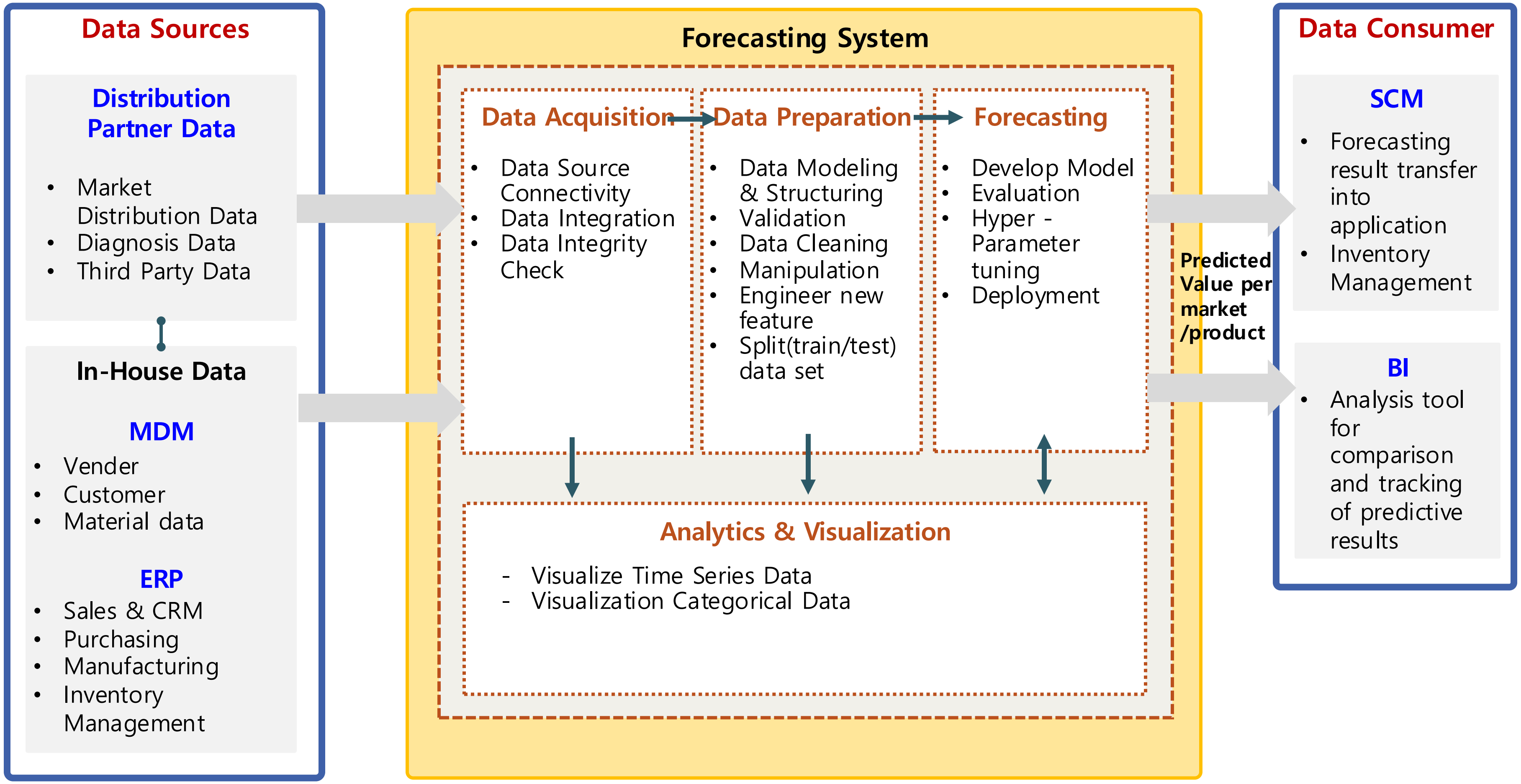
Reflections

- Able to predict whether a user will churn, but not when they will churn
- Considering survival analysis to model churn timing (Cox Regression, Survival Random Forest)

	Early Warning System for Customer Churn	
1	**Business pain-point**	Problem: The company lacks proper churn management, leading to potential revenue loss. ➢ **Develop a churn prevention model to identify at-risk customers and provide actionable insights for retention.**
2	**Project Overview**	Utilize existing customer data to predict churn probability. ➢ **Generate reports on at-risk customers for marketing and sales teams to take proactive retention actions.**
3	**Data Collection**	Customer Data • Feature Types: ➢ Demographics: Gender, age, marital status, and so on ➢ Financials: Income, retirement status ➢ Other Behavioral & Policy Data (Details confidential) ➢ Target: Churn or Not (Binary Classification) - Churned Customer → Has an End-date in their policy
4	**Data Preparation & Exploratory Data Analysis**	• **Highly Imbalanced Data: Churned customers are significantly fewer than non-churned customers.** ➢ Applied SMOTE, random oversampling/undersampling, or cost-sensitive learning. • Handling Missing Values ➢ Used median for skewed data, mean imputation. ➢ For highly correlated features, built a regression model to estimate missing values. • Skewed Data: Price-related features showed skewness. ➢ Applied log transformation for normalization. • Outlier Detection & Treatment ➢ Used IQR (Interquartile Range) method for outlier handling.

	Early Warning System for Customer Churn	
5	**Feature Engineering**	• Create derived features: cancel_customer, dormant, lifetime_value, and so on. • Create interaction Variables: 20 variables ➢ Arithmetic Interaction: e.g. income * age ➢ Conditional Relationship: e.g. income/household_size ➢ Statistical Transformation: e.g. log(income) * education level • Partial PCA: Apply PCA only to highly correlated subgroups related to price, preserving raw features for interpretation. • One-hot encoding: Convert values into numeric values. ➢ Applied Lasso (L1) regression, which forces some coefficients to exactly 0, removing less important features.
6	**Manage Imbalance data**	Consider and experiment with different methods to address data class imbalance: **1) SMOTE (Synthetic Minority Over-sampling Technique)** 2) Random oversample 3) Random undersample 4) Class weight (Cost-sensitive modeling)
7	**Model development & evaluation**	• Prioritizing interpretability over model performance (Understanding churn reasons and root causes is crucial) • Prioritized **churn probability** over estimating user lifetime. (**Classification** vs Cox Regression, Survival Random Forest) • Experimented with and trained various models, including Logistic regression, SVM, Decision Tree, Random Forest, XGBoost, LightGBM ➢ Model Used: XGBoost ➢ Primary Metric: Recall (Focus on minimizing false negatives to avoid missing churned customers). ➢ **Recall: 0.81, Precision: 0.75, F-2(beta) score: 0.80 (More weight on recall than precision)** • To Reduce Overfitting: **PCA, Regularization (L1), VIF (Remove features bigger than 8.0)** • **Due to infrequent updates in customer data, predictions are reported weekly, with model retraining scheduled every month**
8	**Hyper parameter tuning**	Fine-tuned hyperparameters to mitigate overfitting and enhance performances ➢ **max_depth = 6** ➢ **Eta = 0.3** ➢ **Colsample_bytree = 0.7** ➢ **Colsample_bylevel = 0.8**
9	**Reflections and Future Improvements**	• Able to predict whether a user will churn, but not when they will churn • Considering survival analysis to model churn timing (Cox Regression, Survival Random Forest)

✓ **Developed a biosimilar sales forecasting model using SARIMA to optimize production planning, enabling data-driven inventory management.**



	Biosimilar Sales Forecasting Using ARIMA Time Series Analysis	
1	**Business pain-point**	• Lack of demand forecasting for current products prevents optimization of production lines by product type
2	**Project Overview**	• Aim to establish a foundation by developing a baseline demand forecasting model
3	**Data Collection**	• Provided Data Time Range: total 10 years (Monthly herbal medicine data) Regions: Includes US Product Categories: ➢ Dosage Forms ➢ Packaging Types Target Variables: ➢ **Revenue_unit** ➢ Revenue_value ➢ Price
4	**Data Preparation**	• Designed ETL processing to convert wide-format time-series data into long format using pd.melt() for easier analysis. • Baseline modeling: Started with one product as the baseline model. ➢ Trend Analysis: Observed significant differences in sales unit trends across different products. ➢ Conclusion: Requires individualized modeling per product rather than a one-size-fits-all approach.
5	**Exploratory Data Analysis**	• Decomposition Analysis ➢ Compared Original, Seasonal, Trend, and Residual plots to analyze time-series components. • Autocorrelation & Partial Autocorrelation Analysis ➢ Checked ACF/PACF plots to identify dependencies over time. ➢ Observed non-stationarity, indicating the need for transformation. • Differencing for Stationarity ➢ Applied first and second-order differencing to remove trends and stabilize the series.

	<u>Biosimilar Sales Forecasting Using ARIMA Time Series Analysis</u>	

6	**Model selection**	• Decision: Time-Series Analysis Over Regression
		  ➢ Initially considered both regression-based modeling and time-series forecasting
		  ➢ The challenge was determining which features to include (e.g., competitor sales, prescriptions, stock levels, clinical trials, claims data)
		  ➢ Due to the complexity and sequential nature of the data, we chose time-series analysis as the preferred approach
		• Model Selection for Time-Series Analysis
		  • Exponential Smoothing is simple but fails to capture seasonality, as observed in the decomposition analysis.
		  • Since the trend exhibits seasonal patterns, a more advanced model like ARIMA (or SARIMA for seasonality) is required.

7	**Model development & evaluation**	• ACF & Partial ACF Comparison:
		  ➢ Compared MA(q), AR(q), and ARMA(p,q) models.
		  ➢ After second-order differencing, the data exhibited AR(q) characteristics.
		• SARIMA outperformed ARIMA, effectively capturing seasonal trends with a higher R^2 (0.84) and lower AIC (429.78).
		• Effectiveness validation was handed over to the client MLOps team for A/B testing and the deployment.

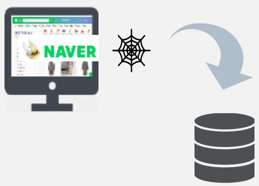
8	**Hyper parameter tuning**	• Hyper parameter tunning & performance comparison:
		  ➢ ARIMA (p:2, d:1, q:2) → AIC: 670, R^2: 0.45
		  ➢ Seasonal ARIMA (SARIMAX) (p:2, d:1, q:2) with Seasonal (P:2, D:1, Q:0, s:12) → AIC: 429.78, R^2: 0.84
		  ➢ Significantly better performance with seasonal adjustment and hyper parameter tuning.

9	**Reflections and Future Improvements**	• Extend the SARIMA model to cover all regions and new pharmaceutical products. • Adapt the model for different market conditions and product-specific trends. • Fine-tune parameters for region-specific and product-level forecasting accuracy.

✓ **Analyze daily news sentiment to provide early notifications, helping stakeholders protect the company's reputation**

✓ **Web crawling**

Web Crawling



Business News

Topic extraction & Article analysis

Rating scores

News Dataset

✓ **Data engineering & model development**

Noise Reduction

Feature Extraction

Text Vectorization

ElasticNet News Sentiment Analysis

Sentiment Analysis Results

	News Sentiment Analysis System for IR Strategy	
1	**Business pain-point**	• Due to high stock price volatility, the company faced increased risk exposure for stakeholders. This volatility created challenges in maintaining financial stability and required proactive measures to mitigate potential risks.
2	**Project Overview**	• To strengthen risk management and enable proactive decision-making, we identified the need for a real-time news sentiment analysis system.
3	**Data Collection**	• Scraped 15,000 news articles from NAVER News • Collected 30 days of data (March 2018), averaging 500 articles daily. • Selected only business-related keywords (e.g., Samsung, display, and so on).
4	**Data Preparation**	• Sentiment Labeling ➢ Employed domain experts for manual sentiment scoring (1-5 scale) to ensure sentiment classification aligns with LG's perspective, accurately assessing whether news articles have a positive or negative impact on the company. ➢ Used scores as training data for a regression model to predict sentiment • Quality Check ➢ Verified sentiment score accuracy before model training.
5	**Data engineering**	• Morphological Analysis • Korean text requires morphological analysis to break words into meaningful units. • Part-of-Speech (POS) Filtering • Select only nouns, verbs, and adjectives • Stopword Removal & Lemmatization • Remove commonly used words • Convert words to their base forms • Deduplication & Special Character Removal • Remove duplicate words to avoid bias in frequency-based models. • Filter out special characters • TF-IDF Transformation • Convert text into a Document-Term Matrix • min_df=3 → Exclude rare words appearing in less than 2–3 documents. • max_df=0.90 → Remove extremely common words appearing in over 95% of documents.

	News Sentiment Analysis System for IR Strategy	

6	**Model selection**	• Model Selection:
		  • **News titles effectively capture the sentiment and key takeaways of the article.**
		  • Efficiency—processing full articles requires more computational resources.
		  • Deep Learning (LSTM, RNN): Not chosen because news titles are short (length <20), making complex models unnecessary.
		  • Explored reinforcement learning models as well but insufficient data for meaningful reinforcement learning.
		• **Decision: ElasticNetCV (Regularized Regression)**
		  • **Applies penalty terms (L1 & L2 regularization) to handle infrequent words.**
		  ➢ Balances feature sparsity (L1) and weight shrinkage (L2) for better generalization

7	**Model development & evaluation**	• Regression Performance:
		  ➢ R^2: 0.79
		  ➢ MAE: 0.35, MSE: 0.2
		• Adjusting Sentiment Classification Threshold (Conservative Approach)
		  • Reason: Reduces false positives, making the model more risk-averse in sentiment classification.
		  ➢ Predicted sentiment score < 3.8 → Negative
		  ➢ Predicted sentiment score ≥ 3.8 → Positive
		• **Automated batch process runs daily at 4 AM to collect, score, and report the previous day's news sentiment to the IR team for proactive risk management**

8	**Hyper parameter tuning**	• l1_ratio = 0.8
		  • l1_ratio = 1.0 → Pure Lasso Regression (stronger feature selection)
		  • l1_ratio = 0.5 → Balanced L1 & L2 (default)
		  • l1_ratio = 0.0 → Pure Ridge Regression (no feature elimination, only weight shrinkage)

9	**Reflections and Future Improvements**	• Expanding Data Sources:
		  ➢ News titles captured sentiment well, but lacked context compared to full articles.
		  ➢ Limited data scope (30 days) might not fully capture long-term sentiment trends.
		• Real-time Monitoring & Deployment
		  ➢ Develop a real-time dashboard for monitoring sentiment trends

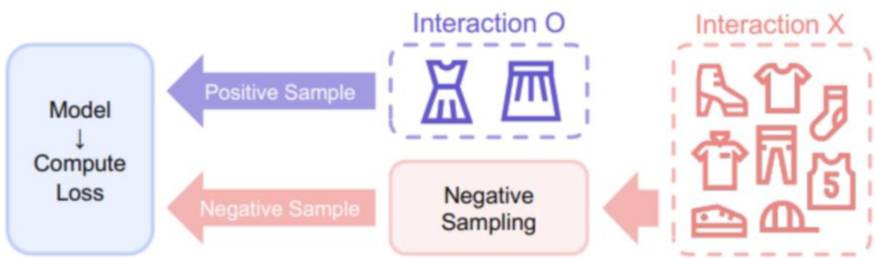
✓ Design and run time zone/age-based A/B tests for female-targeted Facebook ads| 12% improvement in CTR through target segment optimization (Lift 1.12, p-value < 0.05, 30k samples)

No	Contents	Descriptions
1	**Target User**	• Potential customers who interact with ads and marketing c hannels but have not yet converted
2	**Observation**	• Users engaging with the service through Facebook (especially women aged 36-54, active between 12-1 PM) and Email channels, where conversion rates are high
3	**Problem Statement**	• Reduce ad spend on low-conversion channels and reallocate budget to Facebook • Providing more detailed information (images, details, etc.) on Naver, and other channels, similar to email, can improve conversion rates.
4	**Hypothesis**	• H_0: Reducing ad spend on low-conversion channels and reallocating it to Facebook does not result in a statistically significant increase in conversion rate • H_1: Reducing ad spend on low-conversion channels and reallocating it to Facebook leads to a statistically significant increase in conversion rate

No	Contents	Descriptions
6	**Experiment Group**	• Group A (Control Group): Current ad spend distribution re mains unchanged across all channels • Group B (Experimental Group): Increase Facebook ad budg et targeting high-converting segments (women aged 36-54, 12-1 PM).
7	**Experiment Period**	• Base metric: Facebook avg conversion rate • Expected metric: 0.55% • Alpha (Significance): 95% confidence level ($\alpha = 0.05$) • 1-Beta (Power): 80% • Since sample collection was not difficult, there was no need to lower the power. Increasing the power was considered, but the cost was too high to justify. • Minimum Sample size (z-test): around 32,000 people • Period calculation: Required days
8	**Metric**	• Conversion Rate (CTR)
9	**Trade-off**	• Potential loss of customers from other channels, leading to reduced engagement
10	**Andon**	• A significant drop in conversion rate and revenue sustained for more than 3 days.

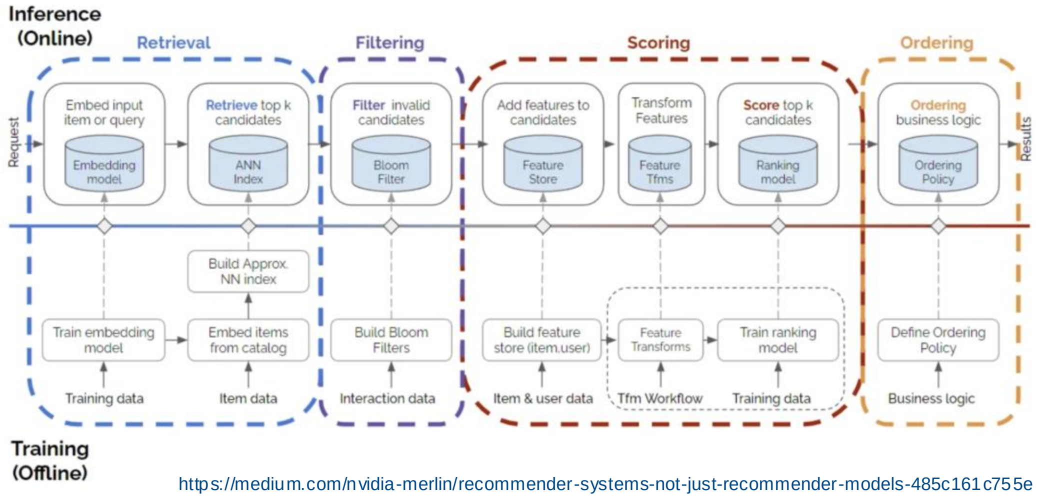
✓ **Developed a 3-stage recommendation system leveraging negative sampling for implicit feedback and ANN-based retrieval, enhancing ranking with ML-based scoring for optimized item recommendations**

✓ Negative sampling for implicit data



- Limited availability of explicit data, such as purchase history or user ratings.
- Apply negative sampling to handle implicit feedback efficiently.
- Items with no interactions are assumed to be irrelevant or less preferred and are sampled as negative instances to optimize model training while reducing computational cost.

✓ 3-stage recommendation system architecture



https://medium.com/nvidia-merlin/recommender-systems-not-just-recommender-models-485c161c755e

- Retrieval: Fetches top-N relevant candidates using an ANN index.
- Filtering: Applies Bloom Filters to remove invalid recommendations.
- Scoring & Ordering:
- Enhances rankings with additional feature transformations.
- Uses a ranking model (e.g., ML-based scoring).

	E-commerce Recommendation Engine: Matrix Factorization with Implicit Feedback and Negative Sampling	
1	**Business pain-point**	• Had customer data available but was not effectively utilizing it to drive business impact
2	**Project Overview**	• Goal: Increase conversion rate (CVR) through a personalized recommendation system. • Initial Approach: Developed an in-house user-based collaborative filtering model themselves. • Issue: Model did not achieve satisfactory performance, requiring further optimization or alternative approaches.
3	**Architecture**	Existing Challenges & Improvements ➢ Challenge 1: Insufficient Training Data a) Explicit data (purchase history, ratings record) was not enough for robust training. b) Solution: Proposed using click history (implicit feedback) to augment training data and improve model performance. ➢ Challenge 2: Limitations of User-Based Collaborative Filtering (CF) a) User-based CF relies only on user-item interactions (ratings, purchases), making it relatively simple. b) Issue: Struggles to capture complex relationships beyond basic similarities. c) Solution: Adopted matrix factorization + feature-based ranking for deeper insights. Proposed a 3-stage Re-Ranking Architecture ➢ Stage 1: SGD-based Matrix Factorization – Generates initial item recommendations. ➢ FAISS Indexer: Enhances efficiency by enabling fast similarity search. ➢ Stage 2: Bloom Filter: Filters out items already sold at the store. ➢ Stage 3: XGBoost Ranker – Refines rankings based on additional features.
4	**Data Collection**	• Since purchase history is insufficient, click data (implicit feedback) is used, and negative sampling is applied by selecting items the user did not interact with. • Negative Sampling Methods for Implicit Data ➢ Random Negative Sampling ➢ **Popularity-Based Negative Sampling: More effective than random sampling in recommendation scenarios**
5	**Data Preparation**	• Sparse Matrix Construction – Converting raw transactional data into a user-click interaction matrix.

	E-commerce Recommendation Engine: Matrix Factorization with Implicit Feedback and Negative Sampling

5	**Model selection & development**	Consider of four recommendation models: • User-based Collaborative Filtering → SGD Matrix Factorization ➢ User-based method: pairwise similarity calculations (computationally expensive), struggles users who have few or no interactions ➢ KNN-based CF: Too simple; performance was suboptimal ➢ **SGD-based Matrix Factorization: captures latent factors, making it effective even with incomplete data.** • Deep Learning-Based Models → Excluded ➢ Lack of organizational expertise in machine learning/deep learning ➢ No available infrastructure to support deep learning deployment.
6	**Model Evaluation**	Stage 1 focuses on retrieval quality (ranking high-relevance items first) using Recall@1000: 0.68 Stage 3 optimizes score calibration for fine-tuning the ranking, where ranking metrics NDCG@10: 0.71 → Top 10 items
7	**A/B test**	• Base Metric: CTR 3.0% • Expected Metric: CTR 3.5% (Around 20.0% improvement) • Controlled Conditions ➢ dd • Statistical Parameters: ➢ Power (1-β): 80% ➢ Significance Level (α): 5% ➢ Minimum Sample Size: 5,295 people (calculated using z-test formula). ➢ Experiment period: daily user: 300 people, required days: 17.7 days *2 group = 35.5days • A/B Test Results: ➢ Lift & p-value Analysis: Confirmed statistical significance of improvements. ➢ **CTR improved by 20% (3.0% → 3.6%), with evaluations conducted across OS types and item categories for deeper insights.** ➢ Business Impact: Demonstrates quantifiable growth driven by optimized product recommendations. ➢ Validation: Confirms positive business impact from strategic recommendation enhancements.

Expected Values on the Continuous Intention to Use IoT Products from the Perspective of Expectation-Confirmation Theory

Master of Science in Information Management

Minseok Oh

KAIST

https://nbviewer.org/github/whommso/ddaf/blob/main/Publication%20Thesis%20KAIST.pdf

	Expected Values on the Continuous Intention to Use IoT Products from the Perspective of Expectation-Confirmation Theory	
1	**Objective**	• Identify gaps between user expectations and actual experience with IoT devices. • Detect pain points and improvement areas to develop user-driven product strategies.
2	**Methodology**	• Structural Equation Modeling (SEM) ➢ Models latent variables to understand how different factors influence user satisfaction. • Causal Analysis ➢ Examines independent-dependent variable relationships to establish causality between user expectations and actual usage experience. • Exploratory Research ➢ Conducts data-driven hypothesis generation to uncover key insights on IoT adoption and usability.
3	**Data collection**	• User Data (Demographics & Behavior) ➢ Collected demographic and usage data: a) Basic Information: Sex, age, income, region, household size, marital status, children. b) IoT Experience: Prior exposure to IoT products, average daily usage • Survey Data (User Experience & Perception) a) Designed 50 survey questions based on previous research, with 4 questions per key variable. b) Key Variables for Analysis: Perceived Manageability, Scalability, Entertainment Value, Reliability, Compensability, Expectation-Performance Alignment, Perceived Usefulness, Perceived Ease of Use, Social Influence, Satisfaction, Intention to Continue Use
4	**Exploratory Data Analysis**	• Reference: document page p.12~14 ➢ https://nbviewer.org/github/whommso/ddaf/blob/main/Publication%20Thesis%20KAIST.pdf
5	**Research model**	• Reference: document page p.8 ➢ https://nbviewer.org/github/whommso/ddaf/blob/main/Publication%20Thesis%20KAIST.pdf

	Expected Values on the Continuous Intention to Use IoT Products from the Perspective of Expectation-Confirmation Theory	
5	**Feature engineering & Assessment**	• Reliability Assessment: ➢ Cronbach's Alpha (CA) ≥ 0.7, CR, and AVE confirm internal consistency & validity. • Correlation Analysis: ➢ All coefficients < 0.85 ensure construct distinctiveness. ➢ Diagonal AVE's square root > correlations confirms discriminant validity. • Model Explanatory Power (R^2): ➢ $R^2 > 0.26$ indicates strong predictive power for dependent variables.
6	**Evaluation**	• Model Construction ➢ Defined path relationships between independent and dependent variables. ➢ Mapped survey scores to respective variables. • PLS-SEM Execution ➢ Path coefficient calculation to quantify relationships. ➢ Bootstrapping (400 resamples) to test statistical significance. ➢ P-value analysis: a. $p < 0.05$ → Reject H_0 (independent variable has a significant impact). b. $p > 0.05$ → Fail to reject H_1 (no significant relationship found). • Interpretation of Results ➢ If H_0 is rejected, e.g., Manageability significantly influences Agreement of Expectation. ➢ If H_0 is not rejected, the factor does not have a statistically significant effect.
7	**Result**	• Income Level: ≥50M KRW → Manageability & Compensability had no impact on satisfaction or continued use. • Household Size: ➢ 1-2 person → Manageability & Entertainingness not significant. ➢ 3+ person → Significant impact on satisfaction & continued use. • Regional Differences: ➢ Metro → Manageability was a dissatisfaction factor. ➢ Non-Metro → Compensability was valued for time & cost savings. • Core Drivers: ➢ Perceived Usefulness → Significant for ≤50M KRW & 3+ person households. ➢ Perceived Ease → Strong positive impact on satisfaction & continued use.

https://github.com/scottmsoh/ref_deep_leraning/tree/main/LLM_chatbot_SCU

Research & Planning

- Define project scope and objectives.
- Analyze existing tools and frameworks for QA and sentiment analysis
- Collect and preproces
s financial news datasets

Core Development

- Implement QA syste
m using sLLM.
- Develop sentiment an
alysis pipeline with fine-tuned models.
- Integrate both compo
nents into one backend system.

Frontend Development

- Design the interface UI for querying and vie
wing results.
- Enable real-time user interaction for QA and s
entiment insights.

Testing & Optimization

- Conduct system testing for accuracy and performance.
- Optimize response time
s and ensure scalability.
- Gather user feedback to debug issues and further refine the dashboard.

End of Document