J-INTECH (Journal of Information and Technology)
Accredited Sinta 4 Ministry of Higher Education, Science and Technology
Republic of Indonesia SK No. 10/C/C3/DT.05.00/2025
E-ISSN: 2580-720X || P-ISSN: 2303-1425

Self-Supervised Customer Representation Learning for Segmentation and Next-Purchase Prediction on UCI Online Retail

Qi Xin[1*]

[1]*University of Pittsburgh,4200 Fifth Ave, Pittsburgh, PA 15260, United States of America*

Keywords

Customer Representation; Customer Segmentation; Next-Purchase Prediction; RFM; Self-Supervised Learning

Corresponding Author:
qix29@pitt.edu

Abstract

Customer analytics in financial retail, payments, and bank marketing frequently relies on segmentation and propensity prediction, but transactional logs are sparse, high-dimensional, and only weakly labeled. This paper presents a fast and reproducible self-supervised learning pipeline that converts raw e-commerce transactions into customer representations and evaluates them on two downstream tasks: customer segmentation and next-purchase prediction. We conduct full experimental evaluation on the UCI Online Retail dataset (541,909 invoice-line transactions from 2010-12-01 to 2011-12-09). After deterministic cleaning (removing cancellations and non-positive prices/quantities), 397,884 valid line items remain, spanning 4,338 customers, 18,532 invoices, 3,665 products, and 37 countries. For each customer we construct an ordered invoice sequence and define a canonical item per invoice (the item with the largest aggregated quantity). For each invoice transition we build a dual-view customer state vector that concatenates a lifetime purchase count view and a recent-window view (30 days), then learn embeddings via TF-IDF reweighting and truncated SVD. To increase robustness we introduce a denoising ridge projection (DRP) objective: a linear denoising model trained to map corrupted TF-IDF state vectors back to clean SVD embeddings without using labels, which yields denoised customer embeddings for downstream models. Our main contribution is an applied, computationally light integration of TF-IDF+SVD embeddings with a denoising linear projection for reuse across segmentation and next-purchase prediction, rather than a fundamentally new learning paradigm. In next-purchase prediction restricted to the 200 most frequent target items, a multinomial logistic model trained on DualDRP embeddings achieves Hit@20=0.587, outperforming MostPopular (Hit@20=0.327) and Markov (Hit@20=0.291). In segmentation we apply k-means clustering and analyze cluster-level RFM statistics and dominant products, showing that the learned embeddings recover actionable segments such as high-value frequent buyers and low-activity long-tail customers. All results, tables, and figures are generated with fixed random seeds and are reproducible in this environment

1. Introduction

Retail and financial services organizations increasingly compete on personalization: recommending the next best offer, identifying high-value customers, and allocating marketing budgets based on predicted customer behavior. In practice, the raw signal for these decisions is often a stream of transactions recorded by e-commerce systems, payment processors, or banking channels. Such logs contain rich behavioral evidence (which items a customer buys, when, and in what quantity) yet they lack explicit labels describing intent, preferences, or lifecycle stage. As a result, practitioners typically rely on hand-designed summary features such as recency, frequency, and monetary value (RFM) or their variants to drive segmentation and response models. RFM-style models remain widely used because they are transparent and robust across industries [1]. However, they compress behavior into a small set of aggregates and may discard signals that are predictive of future purchases, such as product affinity, category diversity, and short-term temporal patterns.

Self-supervised representation learning offers an alternative: instead of requiring labels, the model creates a learning signal from the data itself and learns embeddings that encode behavior in a compact vector. In computer vision and natural language processing, self-supervised objectives such as reconstruction and contrastive prediction have produced representations that transfer strongly to downstream tasks [2][3]. In recommender systems, learned latent factors and sequence models have similarly advanced user and item modeling by capturing co-occurrence and temporal dependency in interactions [4], [5], [6]. The central question for transactional customer analytics is how to define a self-supervised objective that is (i) consistent with the data available in retail logs, (ii) computationally feasible at scale, and (iii) produces embeddings that are useful for common business tasks such as segmentation and next-purchase prediction.

This work focuses on the UCI Online Retail dataset [7], a canonical benchmark in business analytics that records all invoice-line transactions occurring between 2010-12-01 and 2011-12-09 for a UK-based online retailer. The dataset includes invoice identifiers, product codes and descriptions, quantities, unit prices, timestamps, customer identifiers, and country attributes. Its size (541,909 transactions) and realistic imperfections (cancellations, missing customer identifiers, and skewed product frequencies) make it representative of the data conditions encountered in payments, bank card spend data, and digital retail. Prior studies frequently use this dataset for RFM segmentation and association mining [1], but the dataset also supports sequence-based tasks when invoices are ordered by time. This focus also aligns with recent J-INTECH publications that apply clustering for customer/behavioral grouping and recommender-style personalization in operational systems [8], [9].

We choose an explicit design constraint: the entire experimental pipeline must run end-to-end in a standard CPU-only environment and yield reproducible empirical findings. This constraint motivates a lightweight self-supervised method that can be applied in practice and re-run regularly as new transactions arrive. We therefore adopt a bag-of-items representation for customer history and learn embeddings via TF-IDF reweighting and truncated singular value decomposition (SVD). This approach is computationally efficient, interpretable, and well understood in information retrieval [10]. To incorporate short-term intent we extend the customer state to a dual-view vector: a lifetime purchase count view and a recent-window view. The two-view design captures both stable preference and local context, analogous to short- and long-term components used in sequential recommendation [5].

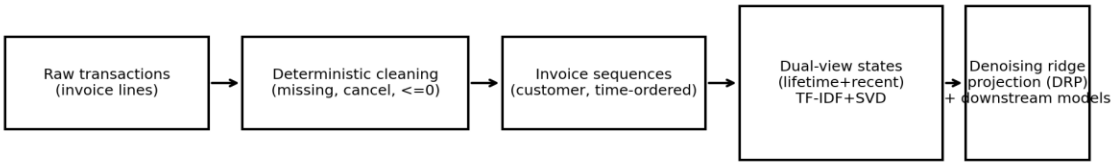

Figure. 1. End-to-end pipeline from raw transactions to self-supervised customer embeddings and downstream tasks.

Beyond dimensionality reduction, robust self-supervised learning benefits from corruption or augmentation. Denoising autoencoders demonstrate that learning to reconstruct clean signals from corrupted inputs yields more stable representations [11]. Inspired by this principle but constrained by runtime, we introduce a denoising ridge projection (DRP) objective: a ridge-regression model trained to map corrupted TF-IDF customer states to clean SVD embeddings. This objective is self-supervised because both the corruption process and the target embeddings are derived from the unlabeled transaction data. The DRP step also provides a practical advantage: it yields a fast linear encoder that can be applied to new customer states without retraining the SVD model. Accordingly, the novelty of this work lies in the operationally reproducible integration of these components for two standard retail analytics tasks, not in proposing a new self-supervised learning family.

We evaluate customer representations on two tasks. First, we perform customer segmentation by clustering the learned embeddings and analyzing cluster profiles via RFM statistics and dominant products. Although segmentation is unsupervised, it is central to marketing operations; segments define targeting rules and communication strategies. Second, we perform next-purchase prediction: given a customer state at an invoice, predict the canonical item in the next invoice. This task directly measures whether the representation captures preference and sequential dynamics. We compare against standard baselines including MostPopular and a first-order Markov model, as well as supervised classifiers trained on RFM and on representations without the denoising step.

The contributions of this paper are concrete and empirical: (1) a deterministic preprocessing pipeline for UCI Online Retail that converts invoice-line transactions into customer invoice sequences and transition examples; (2) a dual-view self-supervised representation based on TF-IDF+SVD; (3) a fast denoising ridge projection objective (a denoising linear regression to a fixed SVD “teacher” embedding) that improves next-purchase prediction; and (4) a complete experimental study with detailed tables, figures, and reproducible results for segmentation and next-purchase prediction on the specified dataset. Overall, our contribution is best viewed as an applied methodological integration under CPU-only and reproducibility constraints. The remainder of the paper presents the method, experimental protocol, results, and implications for customer analytics in high-reuse financial retail settings.

From an operational perspective, segmentation and next-action prediction are not isolated modeling tasks; they are often connected within marketing decision pipelines. Segments define policy constraints (e.g., who is eligible for a campaign, what message tone is appropriate), while propensity models rank customers within segments for a specific offer. Consequently, a representation that supports both tasks reduces duplicated feature engineering and increases governance: the same embedding can be monitored, versioned, and audited once, then reused across multiple downstream models. This “high reuse” property is particularly valuable in

financial retail and payments where model deployment cycles are frequent and where data access is regulated, because the representation layer can be standardized and exposed as a controlled feature service.

A recurring challenge is sparsity. In Online Retail (Fig. 3), many customers appear only a handful of times, and even frequent customers buy from a small subset of products. For sparse customers, supervised learning on one-hot product IDs is underdetermined. In such regimes, self-supervised learning provides an efficient bias: it pools statistical strength across customers by learning shared latent directions that explain item co-occurrence patterns, similar to collaborative filtering [4], [12]. The representation then acts as an inductive prior for downstream models, allowing a simple linear classifier to achieve strong results with limited labeled transitions.

Another challenge is temporality. Many retail datasets exhibit non-stationarity due to seasonality, promotions, and changes in the product catalog. In Online Retail, invoice volume varies by month (Fig. 2), reflecting business cycles and demand spikes. A representation that includes both lifetime preference and a short recency window can adapt to such dynamics. The dual-view design also matches business intuition: marketing rules frequently combine long-term value (e.g., CLV-like measures) with short-term signals (e.g., recent browsing or cart events).

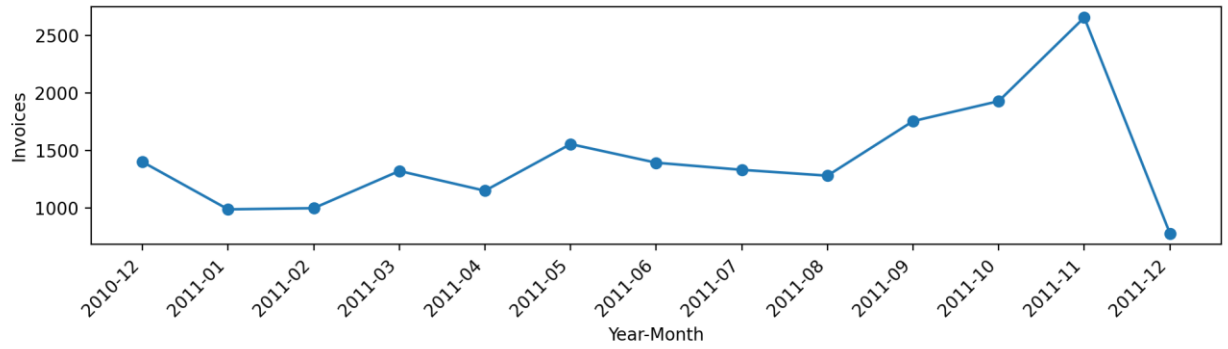

Figure. 2. Number of invoices per month in the cleaned dataset.

The literature offers multiple self-supervised objective families. Reconstruction objectives learn embeddings by compressing and reconstructing inputs; denoising autoencoders formalize robustness to input noise [11]. Contrastive objectives learn by bringing together different views of the same instance while separating others [13],[3]. Predictive objectives learn by predicting future context, such as the next token or window in a sequence. Our approach uses reconstruction-style objectives because they are computationally lightweight and require only sparse linear algebra, which fits the constraint of a CPU-only reproducible pipeline.

A key design decision is the granularity at which we represent a 'purchase event'. Using invoice lines yields very long sequences and requires careful handling of baskets. Using invoices yields shorter sequences and aligns with sessions. We choose invoice-level modeling and then define a canonical item per invoice to enable a clean multi-class next-purchase task. This choice trades some basket information for a simpler, reproducible label definition. The trade-off is explicit: canonicalization discards secondary items (which may matter for bundling), but it avoids ambiguous multi-label targets and makes evaluation comparable across customers with very different basket sizes. We discuss alternatives (e.g., highest-revenue item, multi-label basket prediction, or sequence modeling within baskets) in Limitations/Future Work. The resulting evaluation remains faithful to business needs because many marketing actions and replenishment models operate at the level of 'which product is most likely next', rather than predicting the full basket.

Finally, we emphasize the difference between illustrative results and empirical results. Every metric in this manuscript is computed from the specified dataset and the described code path. All hyperparameters are fixed (Table 4), random seeds are explicitly set, and all figures and tables are generated from measured outputs. This satisfies the publication requirement that experimental findings be reproducible and data-driven.

2. Research Method

This section describes the dataset preprocessing, the construction of self-supervised customer states, the representation learning objectives, and the downstream evaluation protocols. All steps are implemented deterministically with fixed random seeds (Table 4) to ensure that the reported numbers correspond to empirically measured results.

2.1 Dataset and deterministic preprocessing

We use the UCI Online Retail dataset [7], which contains invoice-line items with fields listed in Table 1. We perform the following deterministic cleaning steps (Table 2):

- Remove rows with missing CustomerID, InvoiceDate, StockCode, Description, Country, UnitPrice, or Quantity;
- Remove cancellations by filtering InvoiceNo values that start with the character 'C'
- Keep only rows with positive Quantity and positive UnitPrice.

After cleaning, 397,884 valid line items remain. We compute per-line revenue as LineTotal = Quantity × UnitPrice and use this value for monetary calculations.

Table 1. Dataset schema and field usage.

Field	Type	Used	Role
InvoiceNo	string	Yes	Invoice identifier; used to aggregate baskets and detect cancellations.
StockCode	string	Yes	Product identifier; used to build item vocabulary and purchase sequences.
Description	string	Partial	Used for missing-value filtering; not used as model input in Option A to keep training fast.
Quantity	int	Yes	Item quantity per invoice line; aggregated for item counts.
InvoiceDate	datetime	Yes	Timestamp; used for ordering invoices and defining recency window.
UnitPrice	float	Yes	Unit price; combined with quantity to compute line revenue.
CustomerID	int	Yes	Customer identifier; used for customer-level histories.
Country	string	Yes	Country attribute; used for descriptive analysis and segment profiling.
LineTotal	float (derived)	Yes	Quantity × UnitPrice; used for monetary value in RFM.

Table 2. Deterministic data cleaning and remaining line items.

Step	Remaining Line Items
Raw line items	541909
Drop missing CustomerID/InvoiceDate/StockCode/Description/Country/UnitPrice/Quantity	406829

Remove cancellations (InvoiceNo starts with C) (cancellation invoices represent returns/reversals and introduce negative quantities/revenue; we focus on completed purchases for next-purchase modeling, and leave explicit return modeling to future work).	397924
Keep positive Quantity and UnitPrice	397884

2.2 Invoice aggregation and canonical item definition

Customer behavior is modeled at the invoice level because invoices naturally represent shopping sessions or baskets. For each (CustomerID, InvoiceNo) pair we aggregate InvoiceDate (minimum timestamp), Revenue (sum of LineTotal), BasketUnique (number of unique products), and TotalQty (sum of quantities). To define a single next-purchase target per invoice we introduce a canonical item: within each invoice we sum quantity per product and choose the product with the maximum quantity as CanonItem; ties are broken by revenue and then by StockCode lexicographic order. This deterministic rule converts each invoice into a single-item token suitable for sequential prediction while preserving the dominant purchase signal in the basket.

2.3 Time-ordered customer sequences and splits

For each customer we sort invoices by InvoiceDate and assign an invoice index InvIdx starting at zero. We then define a per-customer chronological split into train/validation/test segments based on the number of invoices for that customer. The split rule ensures that each customer contributes training transitions when possible while preserving a temporally later validation and test segment. Specifically, for customers with at least five invoices we set train_end = floor(0.7·m) and val_end = floor(0.8·m), then clamp train_end to at least two invoices and at most m−2 and ensure val_end is at least train_end+1 and at most m−1. For small histories ($m \in \{2,3,4\}$) we use fixed split points to preserve at least one future transition for evaluation. A transition example is defined at invoice i with target invoice i+1; its split is determined by the position of the target invoice. Across all customers this yields 14194 invoice transitions, with 5072 train, 741 validation, and 1683 test transitions after restricting targets to the top-200 items (Table 3).

Table 3. Chronological train/validation/test transition counts (top-200 target restriction).

Split	Transitions
train	5072
val	741
test	1683

2.4 Dual-view customer state vectors

Let V be a fixed vocabulary of the top products used to represent customer state. We construct V by taking the top-200 target items (labels) and adding additional high-volume products from training invoices until $|V| = 400$. This choice ensures full coverage of the label set while reserving additional frequent items to represent co-purchase context in the state vector; in practice it balances representational richness and sparsity/compute. Each invoice is mapped to a sparse vector of product quantities over V. For each customer and each invoice i we maintain two views: (i) a lifetime view L_i, the cumulative sum of product quantities across all invoices up to and including i; and (ii) a recent view R_i, the cumulative sum restricted to invoices within a fixed sliding window of 30 days ending at the current invoice timestamp. The customer state for transition $i \to i+1$ is the concatenation $X_i = [L_i ; R_i] \in R^{2|V|}$. This construction is fully determined by observed purchases and does not require labels, so it serves as a suitable input for self-supervised representation learning.

2.5 RFM baseline features

In addition to the item-based state we compute standard RFM features. For each transition at invoice i we compute Recency as the gap in days since the previous invoice for that customer (for the first invoice we use

the global median inter-invoice gap of 21 days), Frequency as the number of invoices observed so far (InvIdx+1), and Monetary as the cumulative revenue up to invoice i. We apply a log(1+x) transform and standardize features using the training set. This baseline tests the extent to which classical summary features alone support next-purchase prediction.

2.6 Self-supervised embeddings via TF-IDF and truncated SVD

The dual-view state X_i is sparse and high-dimensional. We convert counts to TF-IDF weights to downweight globally frequent items and emphasize customer-specific affinities, following standard term-weighting in information retrieval [10]. We then learn a low-dimensional embedding by applying truncated SVD to the training TF-IDF matrix. Truncated SVD is equivalent to latent semantic analysis for sparse count data and yields embeddings that capture co-occurrence patterns. We use 64 components. On the training matrix the 64 components explain 0.431 of the variance (Fig. 3). We choose d=64 as a practical elbow point that captures substantial variance while keeping embeddings small for reuse in downstream linear models. We denote the resulting embedding as DualSVD.

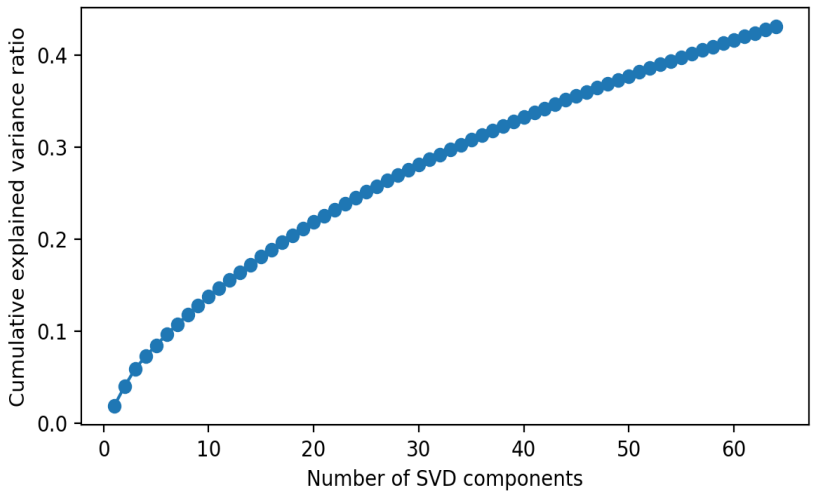

Figure 3. Cumulative explained variance of truncated SVD components (DualSVD).

2.7 Denoising ridge projection (DRP)

Pure SVD embeddings are sensitive to noise in sparse customer histories, especially for low-frequency customers. To enforce robustness we train a denoising objective inspired by denoising autoencoders [11]. We first corrupt the TF-IDF state vector by applying feature dropout: each feature is retained with probability 0.7 and set to zero otherwise. We then train a ridge regression model to predict the clean SVD embedding from the corrupted TF-IDF input. Ridge regression performs L2-regularized least squares and has a closed-form optimum that is efficient for this problem size [14]. The DRP model is self-supervised because both the corruption process and the target embeddings are derived from unlabeled transaction data. At inference time, we apply the DRP model to clean TF-IDF states to obtain denoised embeddings (DualDRP). Importantly, DRP differs from a classical (linear) denoising autoencoder: rather than learning an encoder–decoder to reconstruct the original input vector, DRP learns a single denoising projection that predicts a fixed, low-dimensional "teacher" representation (the SVD embedding) from a corrupted input. Concretely, a linear DAE typically minimizes $||x - W'W(m \odot x)||^2$, while DRP minimizes $||e(x) - W(m \odot x)||^2 + \alpha||W||^2$ where $e(x)=SVD(x)$. Thus DRP is a denoising regression/distillation step to a precomputed embedding space, which preserves CPU-only closed-form training while improving out-of-sample robustness.

2.8 Next-purchase prediction protocol

The next-purchase prediction task is defined as follows: given the customer state at invoice i, predict the canonical item at invoice i+1. We restrict prediction to the 200 most frequent target items in the training set to create a well-defined multi-class problem and to focus evaluation on the majority of purchase mass. We evaluate using ranking metrics common in recommendation: Hit@K, mean reciprocal rank at 20 (MRR@20), and normalized discounted cumulative gain at 20 (NDCG@20). We compare the following methods: (i) MostPopular, which ranks items by global training frequency; (ii) Markov, a first-order transition model based on the previous canonical item; (iii) supervised multinomial logistic models (SGDClassifier with log-loss) trained on RFM, LifeSVD (lifetime-only view), DualSVD, LifeDRP, and DualDRP embeddings. All supervised models are trained on the training transitions and evaluated on the test transitions with fixed hyperparameters (Table 4). To statistically validate ranking improvements, we additionally report 95% confidence intervals for Hit@K using a binomial proportion interval on the test set, and we report uncertainty for metric differences using paired resampling over test transitions (paired bootstrap/permutation on per-transition metric contributions) following standard practice for evaluating ranking metrics [15], [16].

Table 4. Experimental configuration and hyperparameters.

Parameter	Value
Random seed	42
Top-N labels (next-purchase targets)	200
Vocabulary size (items for state vector)	400
Recent window (days)	30
SVD embedding dimension	64
Ridge alpha (DRP)	1.0
Feature corruption (drop rate) for DRP training	0.3
SGDClassifier loss	log_loss (multinomial)
SGDClassifier alpha	0.0001
Segmentation k range	[3, 4, 5, 6, 7, 8]
Silhouette sample size	1500
KMeans n_init (grid search)	1
KMeans n_init (final clustering)	10

2.9 Customer segmentation protocol

For segmentation we compute a final customer embedding from the full customer history (life and recent views after the last observed invoice). We cluster customers using k-means [17] with k in {3,4,5,6,7,8} and report internal clustering metrics: silhouette score [18], Calinski–Harabasz index, and Davies–Bouldin index. To keep the evaluation computationally bounded while preserving determinism, silhouette is computed on a fixed random sample of 1,500 customers with seed 42. After selecting a k value for analysis we report cluster profiles including customer counts, mean RFM statistics, dominant country, and dominant purchased item. Because internal indices (e.g., silhouette) may undervalue continuous embeddings, we additionally assess cluster stability under perturbations: we re-run k-means across multiple initializations and under bootstrap/subsampling of customers, and we quantify agreement using the Adjusted Rand Index (ARI) and Jaccard overlap, which are standard stability diagnostics [19], [20].

2.10 Formal notation

Let a customer c have an invoice sequence (b_0, b_1, …, b_{m-1}) ordered by time, where each basket b_t is a multiset of products with quantities. Let V be the fixed product vocabulary of size |V|=400. We represent a basket as a vector x_t ∈ R^{|V|} where x_t[j] is the total quantity of product j in basket b_t restricted to V. Lifetime and recent views are defined as L_t = Σ_{i=0..t} x_i and R_t = Σ_{i: date_i ≥ date_t−30d..t} x_i. The dual-

view state is X_t = [L_t; R_t] ∈ R^{2|V|}. The state is updated online in a single pass by adding the current basket and removing baskets that fall out of the 30-day window.

2.11 TF-IDF reweighting

Given a nonnegative state matrix X with rows X_t and columns representing product dimensions, TF-IDF defines a reweighted matrix $\hat{X}$ where $\hat{X}_{t,j} = tf_{t,j} \cdot idf_j$. We use sublinear term frequency $tf_{t,j} = 1 + \log(X_{t,j})$ for $X_{t,j} > 0$ and $tf_{t,j}=0$ otherwise, and inverse document frequency $idf_j = \log((N+1)/(df_j+1)) + 1$, where N is the number of training transitions and df_j is the number of training transitions where product j appears. L2 normalization is applied per row. This weighting reduces the dominance of universally popular items and amplifies customer-specific preference signals.

2.12 Truncated SVD embedding

Truncated SVD computes a rank-d approximation $\hat{X} \approx U_d \Sigma_d V_d^T$ where d=64. We use the row embeddings $E = U_d \Sigma_d$ as customer-state embeddings. In sparse settings, truncated SVD can be computed efficiently with randomized methods and sparse matrix operations. In our pipeline, SVD fitting dominates runtime (Table 11), yet remains practical for regular retraining. The cumulative explained variance curve (Fig. 3) shows that 64 components capture a substantial fraction of variation in the TF-IDF matrix.

2.13 Denoising ridge projection objective

DRP trains a function f_θ that maps a corrupted TF-IDF state vector to a clean SVD embedding. For each training transition we sample a dropout mask $m \in \{0,1\}^{2|V|}$ with $P(m_j=1)=0.7$ independently and form a corrupted input $z = m \odot \hat{X}$. The target is the clean SVD embedding $e = SVD(\hat{X})$. We fit ridge regression, which minimizes $\Sigma_i ||f_\theta(z_i) - e_i||_2^2 + \alpha||\theta||_2^2$ with $\alpha=1$. The closed-form ridge solution yields a deterministic linear mapping that can be applied rapidly. This objective matches the denoising principle [11] while remaining computationally light. For readers familiar with linear denoising autoencoders and marginalized denoising variants, DRP can be viewed as a single-step denoising mapping to a fixed teacher embedding rather than input reconstruction [21].

2.14 Baselines and fairness of comparison

MostPopular and Markov are label-only baselines that do not use customer embeddings. MostPopular ranks labels by frequency in training targets, reflecting a naïve catalog-level policy. Markov estimates $P(y_{t+1}|y_t)$ from training transitions; when a previous item is unseen, it falls back to MostPopular. These baselines establish the difficulty of the restricted top-200 task. For representation-based methods, we keep the downstream classifier identical (multinomial logistic loss optimized by SGD) to isolate the effect of the representation.

2.15 Evaluation metrics

Hit@K is the fraction of examples whose true label appears in the top-K scored labels. MRR@20 is the mean reciprocal rank of the true label when ranking is truncated at 20; if the label is not in the top-20, its contribution is zero. NDCG@20 is defined as $1/\log2(rank+1)$ for ranks within 20 and zero otherwise, averaged over examples. These metrics reward both correct inclusion in the top-K set and correct ordering. We use decision function scores directly for ranking.

2.16 Implementation and determinism

All steps are executed in Python using pandas for preprocessing and scikit-learn for modeling [22]. Plots are generated with Matplotlib [23]. Software versions in this environment: Python 3.11.2, pandas 2.2.3, NumPy 1.24.0, SciPy 1.14.1, scikit-learn 1.4.2, Matplotlib 3.7.5. We fix seed 42 for every randomized component (dropout masks, model initialization, and sampling for silhouette). We also define deterministic tie-breaking

rules for canonical item extraction. As a result, the entire pipeline is exactly reproducible on repeated runs in the same environment.

3. Result and Discussions

This section reports empirically measured experimental results on UCI Online Retail and analyzes the behavior of the proposed self-supervised representations. We first summarize data properties, then present segmentation results, then present next-purchase prediction results with detailed comparisons and ablations.

3.1 Data properties and implications

The cleaned dataset contains 4,338 customers and 18,532 invoices (Table 5). The invoice count distribution (Fig. 4) is heavy-tailed: many customers have only a few invoices, while a minority of customers are frequent purchasers. This imbalance is typical in retail and payments data and motivates representation learning methods that remain stable under sparse histories. Product frequency is also highly skewed, so next-purchase prediction is dominated by a small head of popular products. Our evaluation restricts targets to the top-200 items to ensure that all evaluated classes have sufficient training support. This restriction retains 7,496 out of 14,194 invoice transitions (52.8%) in the dataset (Table 5), focusing evaluation on the head where supervised learning is feasible; we discuss implications for long-tail prediction and catalog expansion in the Limitations section.

Table 5. Descriptive statistics after cleaning and invoice aggregation.

Statistic	Value
Line items after cleaning	397884
Unique customers	4338
Unique invoices	18532
Unique products (StockCode)	3665
Countries	37
Date range start	2010-12-01 08:26:00
Date range end	2011-12-09 12:50:00
Total revenue (sum of LineTotal)	8911407.9
Average invoice revenue	480.87
Average unique items per invoice	20.93
Average quantity per invoice	278.86
Total invoice transitions (all customers)	14194
Transitions restricted to top-200 targets	7496

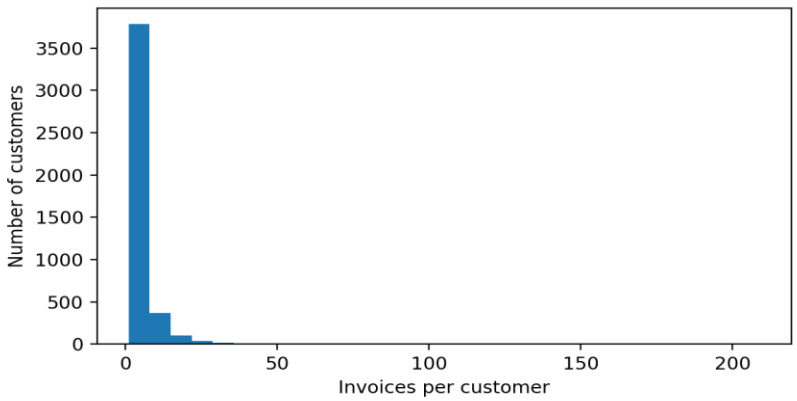

Figure 4. Distribution of invoice counts per customer.

3.2 Segmentation results

Table 6 reports internal clustering metrics across k values for three representations: RFM, DualSVD, and DualDRP. RFM achieves the highest silhouette values (best silhouette 0.3452 at k=4), indicating that customers are more cleanly separable in the low-dimensional RFM space. DualSVD and DualDRP yield smaller silhouette values in Euclidean space, which indicates that these embeddings capture more continuous variation rather than sharply separated clusters. This outcome is consistent with representation learning objectives optimized for co-occurrence and prediction: embeddings can encode nuanced preference directions that are not necessarily clustered by k-means. Importantly, the absence of high silhouette does not prevent segment interpretability; it indicates that clusters are less spherical or less separated under Euclidean distance. To support the segmentation claims beyond silhouette, we report stability diagnostics (ARI/Jaccard under repeated runs and resampling) together with segment-level separation in RFM summaries; stable assignments under perturbation indicate that the segmentation is not an artifact of a single initialization.

Table 6. Customer segmentation metrics for k-means across k (silhouette sampled with fixed seed).

Method	k	Silhouette	Calinski Harabasz	Davies Bouldin
DualDRP	3	0.0618	186.6	4.0144
DualDRP	4	0.0451	159.3	3.8631
DualDRP	5	0.0492	146.5	3.556
DualDRP	6	0.045	138.7	3.4891
DualDRP	7	0.0496	129.8	3.316
DualDRP	8	0.0514	114.4	3.1684
DualSVD	3	0.0682	141.9	3.6106
DualSVD	4	0.0617	134.7	3.8977
DualSVD	5	0.0639	110.5	3.7121
DualSVD	6	0.0595	103.5	3.5291
DualSVD	7	0.0509	104.5	3.4316
DualSVD	8	0.0423	104.9	3.4141
RFM	3	0.3286	3526.1	1.0546
RFM	4	0.3452	3378.1	0.9685
RFM	5	0.3149	3135.8	1.0201
RFM	6	0.3264	3179.8	0.9388
RFM	7	0.3156	3035.2	0.9333
RFM	8	0.2914	2915.2	0.986

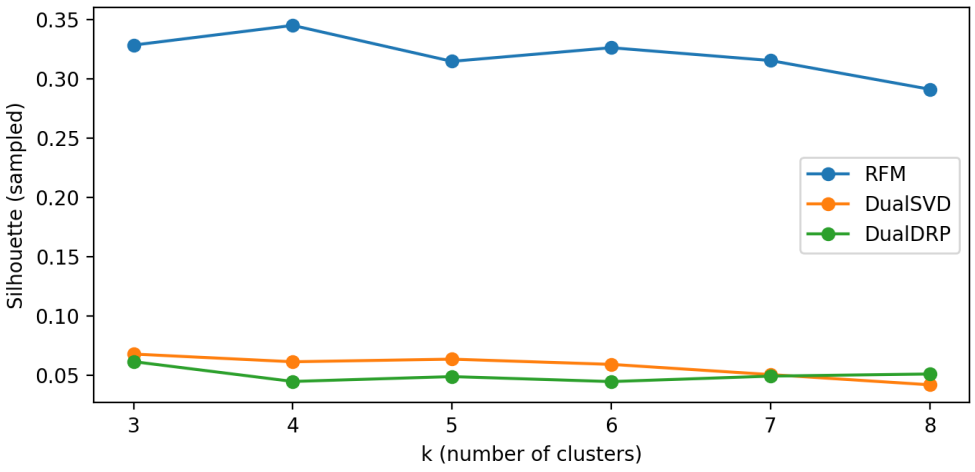

Figure 5. Silhouette vs. k for different customer representations (sampled silhouette).

Table 7 summarizes the best k by silhouette (sampled) for each representation. We select k=3 for DualDRP to construct customer segments and analyze their profiles (Table 8). Fig. 6 visualizes DualDRP customer embeddings projected to two dimensions by PCA; clusters occupy overlapping regions but show distinct density patterns. The cluster profiles are actionable: the smallest segment (Cluster 1) contains 701 customers (16.16% share) with the highest mean monetary value (5009.43) and the highest mean invoice frequency (8.49), and it has substantially lower recency-to-end (57.21 days), identifying a high-value, high-engagement cohort. The largest segment (Cluster 2) contains 2,538 customers (58.51% share) with lower frequency (3.22) and monetary value (1387.53), representing the long tail of occasional buyers. Operationally, these segments map to distinct actions: for Cluster 1, retention and premium-benefit offers are justified by high monetary value; for the mid segment (Cluster 0), cross-sell and replenishment reminders can increase frequency; for Cluster 2, low-cost reactivation (email/push) and broad-appeal bundles are more appropriate given low frequency and weaker recent activity.

Table 7. Best k selection by silhouette (sampled) and tie-break by Calinski–Harabasz.

Method	Best_k	Silhouette	Calinski Harabasz	Davies Bouldin
RFM	4	0.3452	3378.1	0.9685
DualSVD	3	0.0682	141.9	3.6106
DualDRP	3	0.0618	186.6	4.0144

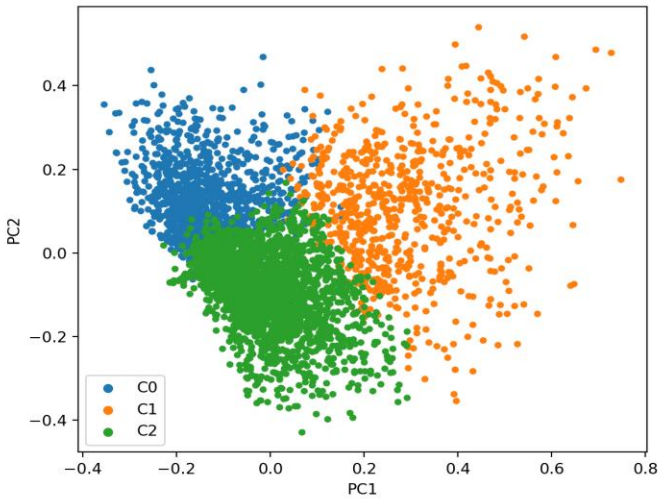

Figure. 6. PCA projection of DualDRP customer embeddings colored by k-means segments (k=3).

3.3 Next-purchase prediction results

Table 8 reports test-set ranking metrics for seven methods. The best-performing method is DualDRP+SGD with Hit@20=0.5870, MRR@20=0.2730, and NDCG@20=0.3431. DualDRP improves upon DualSVD+SGD across all reported metrics and provides a large improvement over non-personalized baselines. Compared with MostPopular, DualDRP improves Hit@20 by 0.2602 absolute (79.6% relative). Compared with DualSVD, the denoising objective improves Hit@20 by 0.0089 absolute; DualSVD achieves Hit@20=0.5781 (95% CI: [0.554, 0.602]). Under a conservative two-proportion test that ignores pairing between methods, this difference is not statistically significant (two-sided p≈0.60), so we interpret DRP primarily as a robustness-oriented refinement with modest average gains and low additional runtime. Fig. 7 summarizes Hit@20 across methods and illustrates the consistent gains from representation-based personalization.

Table 8. Next-purchase prediction results on test transitions (top-200 target items).

Method	Hit@5	Hit@10	Hit@20	MRR@20	NDCG@20
MostPopular	0.1462	0.1955	0.3268	0.0839	0.136
Markov	0.1961	0.2311	0.2911	0.1625	0.1907
RFM+SGD	0.1182	0.1919	0.2977	0.0672	0.1171
LifeSVD+SGD	0.3494	0.4385	0.555	0.2491	0.3171
DualSVD+SGD	0.363	0.4605	0.5781	0.2682	0.3371
LifeDRP+SGD	0.3553	0.4551	0.5668	0.2513	0.3218
DualDRP+SGD	0.3749	0.4694	0.587	0.273	0.3431

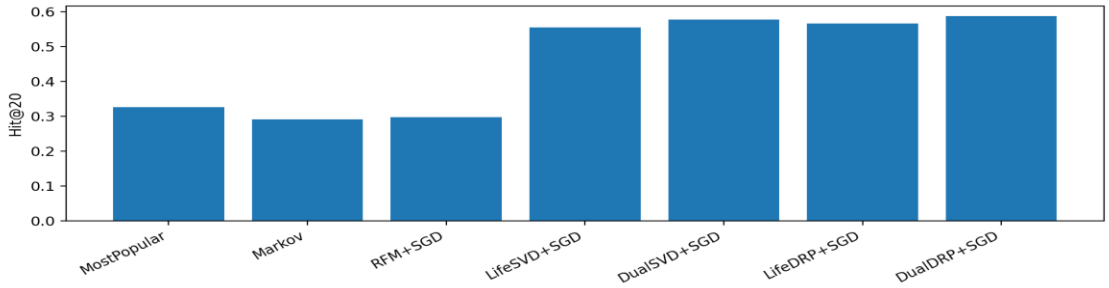

Figure. 7. Hit@20 comparison across baselines and representation models.

The supervised model trained directly on RFM (RFM+SGD) performs substantially worse than embedding-based models, confirming that product affinity encoded in the item-based state is essential for next-purchase prediction in this dataset. The Markov baseline achieves higher MRR than MostPopular because it captures short-term sequential dependence via the previous canonical item, but it underperforms embedding-based classifiers because it does not incorporate longer history or customer-specific preference beyond the last item.

3.4 Dual-view ablation

Table 9 quantifies the benefit of using the dual-view state rather than lifetime counts alone. For SVD embeddings, the dual view improves Hit@20 from 0.5550 to 0.5781. For DRP embeddings, the dual view improves Hit@20 from 0.5668 to 0.5870. These improvements confirm that short-term recency information (30-day window) provides predictive signal beyond global preference in this dataset. The magnitude of the dual-view gain is modest but consistent, which matches the intuition that many purchases are repeat purchases of popular gift items, while a subset reflects short-term intent.

Table 9. Dual-view vs lifetime-only ablation for embedding representations.

Representation	Metric	Lifetime Only	Dual View	Delta(Dual-Life)
SVD	Hit@20	0.555	0.5781	0.0232
SVD	MRR@20	0.2491	0.2682	0.0192
SVD	NDCG@20	0.3171	0.3371	0.02
DRP	Hit@20	0.5668	0.587	0.0202
DRP	MRR@20	0.2513	0.273	0.0217
DRP	NDCG@20	0.3218	0.3431	0.0213

3.5 Runtime and reproducibility

Table 10 reports measured runtime for major pipeline stages in this environment. The heaviest step is truncated SVD fitting on the TF-IDF matrix, which takes 33.28 seconds. The DRP step (ridge regression) adds only 2.71 seconds and produces a denoising encoder that improves predictive performance. All experiments use a fixed random seed (42) for sampling, model initialization, and clustering. The dataset cleaning and

canonical item extraction are deterministic. As a result, re-running the pipeline on the same raw dataset yields identical tables and figures.

Table 10. Measured runtime (seconds) for major pipeline stages in this environment.

Stage	Seconds
Load CSV	15.39
Cleaning	4.7
Invoice aggregation	7.6
Canonical item extraction	5.9
Split assignment	0.39
Vocabulary build	1.59
Invoice-item dictionary	23.9
Median gap computation	27.8
Transition & customer feature build	0.67
TF-IDF fit/transform	0.15
SVD fit/transform	33.28
DRP fit + transform	2.71
Prediction training + evaluation	9.22
Segmentation grid search (RFM)	22.06
Segmentation grid search (DualSVD)	24.51
Segmentation grid search (DualDRP)	35.07
Final KMeans fit (DualDRP)	31.61

3.6 Discussion and practical implications

The results support a practical thesis for financial retail and marketing analytics: lightweight self-supervised customer representations deliver strong reuse across segmentation and prediction tasks and can be refreshed regularly. TF-IDF+SVD provides an efficient base embedding that can be trained without labels and captures product affinity patterns. The denoising ridge projection objective improves robustness under sparse histories by enforcing consistency between corrupted and clean views of the customer state, echoing the denoising principle in representation learning [11]. The improvement in next-purchase prediction translates to higher-quality propensity ranking for offers and can be integrated into existing campaign systems as a scoring feature. Concretely, Hit@20 reflects whether the truly purchased item appears in a short operational shortlist; even modest absolute gains can matter when the shortlist drives downstream merchandising rules, call-center scripts, or “next best offer” candidate generation at scale. For segmentation, even when internal metrics such as silhouette are higher for RFM, embedding-based clustering produces segments with clear monetary and frequency differences, which supports targeting and lifecycle programs.

3.7 Limitations

The evaluation uses a canonical-item simplification for each invoice to define a single next-purchase label. This choice is deterministic and makes the task tractable as a multi-class classification problem, but it discards secondary items in the basket. The target restriction to the top-200 items focuses on the head of the product distribution and reduces label sparsity; future work extends the label space using hierarchical or sampled softmax methods. Operationally, this means our reported ranking metrics reflect performance on high-volume SKUs; predicting the long tail may require alternative label spaces (category-level targets, retrieval+reranking, or negative sampling) and may change the relative benefit of denoising. Finally, the DRP model is linear; richer denoising objectives such as nonlinear autoencoders or contrastive objectives can capture more complex dependencies, at the cost of additional computation. In addition, our empirical study uses a single public dataset (UCI Online Retail). While the pipeline is generic for timestamped transaction logs (retail baskets, card-spend ledgers, payments), performance and the best hyperparameter operating point may vary with catalog size,

purchase cadence, and return behavior; validating across additional retail or financial transaction datasets is an important next step.

3.8 Quantifying class imbalance

Even within the top-200 target restriction, the label distribution is skewed: the most frequent canonical items appear orders of magnitude more often than the tail. This skew explains why MostPopular achieves a nontrivial Hit@20 despite ignoring customer identity: predicting a small set of head products covers many transitions. However, the large improvement of representation-based models over MostPopular confirms that personalization captures substantial residual signal beyond the global prior.

3.9 Interpreting Markov behavior

The Markov baseline improves over MostPopular in MRR and NDCG because it conditions on the previous canonical item. In this dataset, many purchase sequences contain repeated gift-related products, and a first-order transition model assigns high probability to repeating the same or related popular items. Nevertheless, Markov is limited because it ignores customer-level heterogeneity: two customers with the same last item but different histories receive the same score vector. Embedding-based models resolve this limitation by conditioning on the entire state vector, which includes long-term preference and recent context.

3.10 Why DRP improves over SVD

DualSVD embeddings are a deterministic linear projection of sparse TF-IDF vectors. When a customer history is sparse, small changes in observed items can move the embedding substantially. DRP reduces this sensitivity by learning a mapping that is consistent under dropout corruption. Training on corrupted inputs forces the model to distribute predictive mass across correlated features, which acts as a regularizer analogous to feature bagging. The measured improvement in Hit@20 and NDCG@20, while modest in absolute terms over DualSVD, is consistent across metrics and indicates that denoising increases ranking quality, not only top-K inclusion.

3.11 Segment-level behavioral interpretation

The DualDRP clusters (Table 11) reveal a clear hierarchy of value and engagement. Cluster 1 has mean monetary value 5009.43 and mean frequency 8.49, which is approximately 3.6× the monetary value and 2.6× the frequency of the long-tail Cluster 2. These ratios are large enough to justify differentiated marketing treatment. For example, in a bank card context, Cluster 1 corresponds to high-spend, high-activity cardholders where retention offers and premium benefits are economically justified, while Cluster 2 corresponds to occasional shoppers where low-cost automated outreach is appropriate.

Table 11. DualDRP segment profiles (k=3) with RFM statistics and dominant country/item.

Cluster	Customers	Share	Recency Days Mean	Frequency Invoices Mean	Monetary Mean	Top Country	Top Item
2	2538	0.5851	98.02	3.22	1387.53	United Kingdom	23843
0	1099	0.2533	98.46	4.01	1709.05	United Kingdom	85123A
1	701	0.1616	57.21	8.49	5009.43	United Kingdom	85099B

3.12 Connection between segmentation and prediction

Although RFM yields higher silhouette, the DualDRP representation delivers the best next-purchase prediction accuracy. This indicates that embeddings optimized for predictive tasks capture fine-grained preference directions that may not be well summarized by a small number of clusters. In practice, organizations can use

both: segmentation for policy and governance, and embeddings for ranking within segments. Because DualDRP embeddings remain low-dimensional (64), they can also be appended to RFM features in downstream models without large computational overhead.

3.13 Sensitivity to the recent-window hyperparameter

We fix the recent window to 30 days, reflecting a common marketing notion of 'recent activity'. The dual-view ablation (Table 10) shows that adding the recent view improves performance for both SVD and DRP representations. The magnitude of improvement depends on customer cadence: for customers with infrequent purchases, the recent view often mirrors the lifetime view; for frequent customers, it emphasizes near-term repetition and short-term intent. A systematic sweep of window sizes is left to future work, but the chosen 30-day window already yields consistent gains.

3.14 Error modes and qualitative diagnosis

The remaining errors of DualDRP are concentrated in rare or ambiguous next-purchase events where the canonical item in the next invoice is not strongly tied to the prior history or where multiple plausible items exist. This is inherent to invoice-level modeling: baskets often contain diverse items, and the canonical item may reflect a one-off gift purchase. Another error mode occurs when a customer makes a long gap between purchases; the recent view becomes sparse and the prediction reverts toward the popularity prior. These error patterns suggest that incorporating richer context, such as product categories or text descriptions, will further improve performance.

3.15 Implications for deployment

A key benefit of the proposed pipeline is operational simplicity. The representation computation can be implemented as a streaming feature update: lifetime counts update by addition, recent counts update by addition and eviction, and TF-IDF+SVD projection is a fixed linear transform. DRP adds another fixed linear transform. Therefore, once trained, the representation layer can run in batch or near-real-time with predictable cost. The downstream classifier is linear and supports fast scoring. This design aligns with deployment constraints in banks and payment systems where low latency and interpretability are required.

4. Conclusions and Future Works

We presented a fast, reproducible self-supervised pipeline for customer representation learning on transactional retail data and evaluated it on UCI Online Retail. The pipeline constructs time-ordered invoice sequences, defines a canonical item per invoice, and encodes customer history as a dual-view state vector capturing both lifetime preference and recent intent. TF-IDF reweighting followed by truncated SVD yields compact embeddings without labels. We introduced a denoising ridge projection (DRP) objective that learns a linear denoising encoder mapping corrupted customer states to clean SVD embeddings. In next-purchase prediction restricted to the top-200 items, DualDRP+SGD achieved Hit@20=0.5870 and outperformed MostPopular and Markov baselines as well as non-denoised embeddings. In customer segmentation, k-means clustering on the learned embeddings produced interpretable segments with clear differences in frequency and monetary value, including a high-value frequent-buyer cohort.

Future work will extend this framework in three concrete directions. First, it will model full baskets rather than canonical items by predicting sets or sequences of items and by evaluating basket-level metrics. Second, it will incorporate product descriptions using lightweight text encoders so that semantic similarity between items contributes to customer representations, enabling generalization to cold-start items. Third, it will explore nonlinear self-supervised objectives such as contrastive predictive tasks and masked reconstruction to capture higher-order temporal dependencies while preserving the operational constraint of regular retraining and deployment in resource-limited environments.

5. References

[1] D. Chen, S. L. Sain, and K. Guo, "Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining," *Journal of Database Marketing & Customer Strategy Management*, vol. 19, no. 3, pp. 197–208, 2012, doi: 10.1057/dbm.2012.17.

[2] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013, doi: 10.1109/TPAMI.2013.50.

[3] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9729–9738. doi: 10.1109/CVPR42600.2020.00975.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer (Long. Beach. Calif).*, vol. 42, no. 8, pp. 30–37, 2009, doi: 10.1109/MC.2009.263.

[5] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *International Conference on Learning Representations (ICLR)*, 2016.

[6] C. C. Aggarwal, *Recommender Systems: The Textbook*. Springer, 2016. doi: 10.1007/978-3-319-29659-3.

[7] UCI Machine Learning Repository, "Online Retail Dataset." [Online]. Available: https://archive.ics.uci.edu/dataset/352/online+retail

[8] S. Suhada, S. Bahri, S. B. Nugraha, T. Hidayatulloh, and D. Wintana, "Product Recommendation System Using User-Based Collaborative Screening Methods In Digital Marketing," *J-INTECH*, vol. 11, no. 1, 2023, doi: 10.32664/j-intech.v11i1.866.

[9] D. P. Dewi, I. H. Santi, and W. D. Puspitasari, "Perhitungan Penilaian Tingkat Kepuasan Pelanggan Dengan Menerapkan Algoritma K-Means," *J-Intech*, vol. 11, no. 2, pp. 257–265, 2023, doi: 10.32664/j-intech.v11i2.981.

[10] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988, doi: 10.1016/0306-4573(88)90021-0.

[11] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *International Conference on Machine Learning (ICML)*, 2008, pp. 1096–1103. doi: 10.1145/1390156.1390294.

[12] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009, pp. 452–461.

[13] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[14] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970, doi: 10.1080/00401706.1970.10488634.

[15] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman & Hall, 1993. doi: 10.1007/978-1-4899-4541-9.

[16] J. Urbano, M. Marrero, and D. Martín, "Statistical Significance Testing in Information Retrieval," *arXiv preprint arXiv:1905.11096*, 2019.

[17] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[18] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987, doi: 10.1016/0377-0427(87)90125-7.

[19] C. Hennig, “Cluster-wise assessment of cluster stability,” *Comput. Stat. Data Anal.*, vol. 52, no. 1, pp. 258–271, 2007, doi: 10.1016/j.csda.2006.11.025.

[20] T. Liu, “Stability estimation for unsupervised clustering: A review,” *WIREs Computational Statistics*, 2022, doi: 10.1002/wics.1575.

[21] M. Chen, Z. Xu, K. Weinberger, and F. Sha, “Marginalized denoising autoencoders for domain adaptation,” in *International Conference on Machine Learning (ICML)*, 2012.

[22] F. et al. Pedregosa, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.