Diagnosing and Resolving Feature Space Noise in Large-Scale Customer Action Propensity Prediction

Daniel Will George
Daniel George Research
Los Angeles, California
daniel.george@stanford.edu

***Abstract*—Large-scale machine learning pipelines for customer lifetime value (LTV) prediction routinely encounter computational bottlenecks that practitioners misattribute to insufficient infrastructure. We present a systematic investigation of such a bottleneck in a production retail personalization system, where hyperparameter optimization for a multi-class action propensity classifier required over 18 hours on 300,000 training examples—and failed entirely on datasets exceeding 30 million records. Rather than scaling compute, we diagnose the root cause as *feature space noise*: a 330+ dimensional feature space characterized by severe class imbalance (<0.4% positive class prevalence), pervasive inter-feature Pearson correlations exceeding 0.99, and 14 near-zero-variance features contributing no discriminative signal. Using t-SNE and UMAP visualization, Pearson correlation analysis, chi-squared statistical testing, and variance filtering, we reduce the feature space by over 95%—from 330+ to 19 features—while substantially improving class boundary separability. Following feature reduction, equivalent hyperparameter search completed in under 2 hours on a GPU cluster with 1 million training examples. Our work demonstrates that in severely imbalanced, high-dimensional production settings, principled feature space diagnosis yields greater efficiency gains than scaling compute infrastructure, and provides a reusable diagnostic methodology for practitioners facing similar challenges.**

***Index Terms*—feature selection, dimensionality reduction, class imbalance, customer lifetime value, t-SNE, UMAP, hyperparameter optimization, retail personalization, Apache Spark**

I. Introduction

Predicting customer lifetime value (LTV) and behavioral action propensity is a central challenge in large-scale retail personalization. Accurate propensity models power downstream decisions in marketing investment, digital experience design, and customer retention—making model quality and training efficiency jointly important at production scale.

A critical component of modern LTV frameworks is a multi-class action propensity classifier that assigns each member to one of three behavioral states: *inactive*, *active-non-transacting*, or *active-transacting*. Distinguishing these states— particularly identifying the commercially critical minority of actively purchasing members—enables personalized interventions that can meaningfully shift long-term customer value [1].

A persistent challenge in deploying such systems is the computational cost of hyperparameter optimization. In our setting, after feature engineering and enrichment across four digital engagement platforms and multiple temporal lookback windows, the initial feature space comprised over 330 dimensions, combining numerical engagement metrics with one-hot-encoded categorical transaction features. With 300,000 training records and 150 Optuna [2] trials using a RandomForest classifier [3] on a distributed Apache Spark [4] cluster, a single hyperparameter search required over 18 hours to complete. At production scale—datasets exceeding 30 million records—the same pipeline failed to complete a single trial after 16 hours, accompanied by persistent out-of-memory (OOM) errors despite systematic tuning of executor memory, worker count, and cluster configuration parameters.

The natural response to such bottlenecks is to scale infrastructure: increase executor memory, expand cluster workers, or migrate to GPU-accelerated training. We pursued all of these interventions and found none sufficient. This prompted a different diagnostic question: rather than *how can we give the model more resources*, we asked *why does the model need so many resources in the first place?*

The answer, as we demonstrate, lies in the feature space itself. Visualization via t-SNE [5] and UMAP [6] reveals that the 330+ dimensional feature space produces no meaningful class separation. Correlation analysis exposes pervasive near-perfect redundancy across temporal window variants of the same base features. Variance analysis identifies 14 features contributing zero discriminative signal. Through systematic application of these diagnostic tools followed by principled feature selection, we reduce the feature space from 330+ to 19 dimensions and transform an intractable training pipeline into one that completes in under 2 hours.

Contributions.

1) A systematic *feature space diagnostic methodology* combining visualization, correlation analysis, variance filtering, and statistical testing, applicable to any high-dimensional production ML setting.
2) Empirical demonstration that >95% feature reduction *improves* class separability while reducing hyperparameter optimization time from 18+ hours to under 2 hours on significantly larger data.
3) Quantitative characterization of the feature redundancy patterns—temporal window replication, platform-level anti-correlations, zero-variance engagement rates—that systematically inflate dimensionality in retail engagement data.
4) Practical guidance on algorithm selection (tree-based vs. neural network) based on observed feature space

geometry, with implications for production scale and model interpretability.

II. Related Work

A. Feature Selection

Feature selection is a well-studied subproblem in machine learning [7]. Methods are broadly classified as filter methods (statistical scoring independent of the learning algorithm), wrapper methods (using model performance as the selection criterion), and embedded methods (selection as part of training, e.g., tree-based feature importance [3]). In high-dimensional settings with correlated inputs, filter methods—including variance thresholding and correlation analysis [8]—are the most computationally practical first step and are particularly well-suited to the diagnostic phase we describe.

B. Dimensionality Reduction for Visualization

t-SNE [5] is a nonlinear dimensionality reduction technique that preserves local neighborhood structure, revealing cluster separation—or its absence—in high-dimensional data. Unlike PCA, t-SNE handles mixed numerical and one-hot-encoded categorical data well, making it appropriate for our feature space. UMAP [6] offers complementary strengths: it preserves both local and global structure while scaling more efficiently to larger datasets. We use both methods not for representation learning but as diagnostic instruments to assess feature quality prior to model training.

C. Class Imbalance

Severe class imbalance presents well-documented challenges for classification algorithms [9]. When the minority class comprises less than 1% of training data, standard accuracy metrics become misleading and tree-based models must construct extremely deep, complex decision boundaries to capture the positive class—a key contributor to the computational burden we diagnose. Standard remediation strategies include oversampling [9], class weighting, and threshold calibration.

D. Customer Lifetime Value Prediction

LTV prediction has a rich literature spanning probabilistic models [1] and modern ML approaches. The challenge of multi-platform behavioral feature engineering—integrating signals from mobile apps, web, and transactional history across multiple temporal windows—has received comparatively less attention, particularly regarding the computational scalability implications at tens of millions of records.

E. Hyperparameter Optimization at Scale

Optuna [2] provides state-of-the-art hyperparameter optimization through tree-structured Parzen estimators and pruning. However, its efficiency is bounded by per-trial training cost—a cost that scales poorly with feature dimensionality and class imbalance in tree-based models [3], motivating feature-level intervention before optimization.

III. Problem Formulation

We consider a multi-class classification problem over a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector derived from member engagement and transaction history, and $y_i \in \mathcal{Y} = \{\textit{inactive}, \textit{active-non-transacting}, \textit{active-transacting}\}$ is a behavioral state label representing a member's action propensity.

The label space reflects the following behavioral segments:

- **Inactive**: Members with no recent engagement across tracked platforms.
- **Active-non-transacting**: Members actively engaging but not purchasing.
- **Active-transacting**: Members actively engaging and purchasing—the minority class of primary commercial interest.

The dataset contains $N \approx 39.5 \times 10^6$ labeled records drawn from a single geographic region and primary digital platform. Features $\mathbf{x}_i$ are constructed from behavioral engagement signals across four digital platforms (Platforms A, B, C, and D) at five temporal lookback windows (7, 28, 84, 168, and 336 days), yielding an initial feature dimensionality of $d > 330$.

Our primary objective is to identify a reduced feature set $\mathbf{x}'_i \in \mathbb{R}^{d'}$, $d' \ll d$, such that: (1) class boundaries in the reduced space $\mathbb{R}^{d'}$ are more separable than in $\mathbb{R}^d$; (2) hyperparameter optimization over a model trained on $\mathbf{x}'$ is computationally tractable at production scale; and (3) predictive performance is maintained or improved.

IV. Dataset and Exploratory Analysis

A. Data Overview

Features are organized into two types: (1) **Numerical features** (110 in the initial selection), including platform tenure, recency measures (*days_since_inactive*, *days_since_last_purchase*, etc.), engagement rates, and transactional signals (order size, unit count, monetary value, LTV) each replicated across five temporal windows; and (2) **Categorical features** (24 pre-encoding), capturing product category and order channel for first and last transactions, replicated across six temporal windows. One-hot encoding of categorical features expands the total dimensionality beyond 330.

B. Class Distribution

The dataset exhibits extreme class imbalance, illustrated in Figure 1. The majority *inactive* class contains 33,826,365 records; *active-non-transacting* contains 5,503,145 records; and the commercially critical *active-transacting* class contains only 131,833 records—approximately 0.34% of the inactive class and 0.33% of all records.

This imbalance has a direct consequence for tree-based models: to correctly classify the minority class at any reasonable recall threshold, a RandomForestClassifier must construct increasingly deep trees, multiplying the per-trial training cost in hyperparameter search [3].

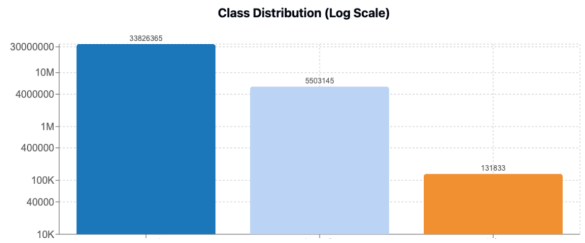


Fig. 1: Class distribution on log scale. *Active-transacting* (orange, $n = 131{,}833$) comprises $< 0.4\%$ of *inactive* (dark blue, $n = 33{,}826{,}365$), imposing a severe imbalance challenge for multi-class classification.

C. Feature Distribution Analysis

Statistical analysis of the 110 numerical features reveals properties relevant to algorithm selection. Critically, *no feature exhibits a normal (Gaussian) distribution*. Of the 110 numerical features, 92 are right-skewed, 18 are left-skewed, and 30 show evidence of bimodal or multimodal structure. High variance is concentrated among monetary, LTV, past demand, order size, and platform tenure features, with maximum values reaching 6×10^6.

These distributional properties have direct algorithmic implications. Tree-based models are robust to feature skewness, as splits are computed on rank order rather than magnitude [3]. Neural networks [10], however, require normalization to prevent gradient instability—an important preprocessing consideration when evaluating GPU-accelerated alternatives.

V. Methodology

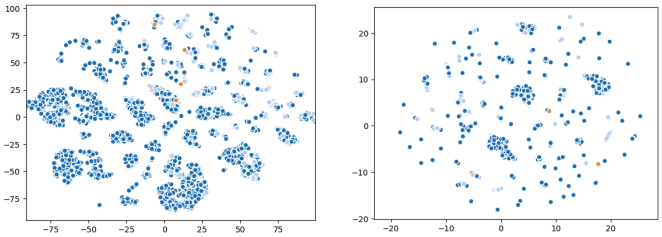
Our feature analysis methodology combines four complementary approaches applied in sequence: (1) dimensionality reduction visualization, (2) pairwise Pearson correlation analysis, (3) variance filtering, and (4) chi-squared statistical testing. Each method diagnoses a different failure mode in the feature space. Together they provide a multi-angle characterization of feature quality before a single model is trained.

A. Dimensionality Reduction Visualization

To qualitatively assess class separability, we apply t-SNE [5] and UMAP [6] to stratified samples drawn to preserve the class distribution. We frame the visualization as a binary classification problem—designating *active-transacting* as the positive class—to simplify interpretation and reduce computation.

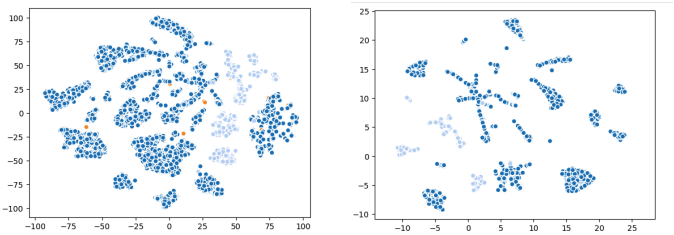
Figure 2 shows the projections of the 330+ feature space. We observe four diagnostic indicators of a problematic feature space: (1) **scattered distribution**—points are widely dispersed with no coherent cluster structure; (2) **class overlap**—positive class points are embedded within dense regions of negative class points; (3) **diffuse boundaries**—even where loose groupings form, boundaries are irregular; and (4) **isolated outliers**—numerous small isolated point groups characteristic of noise rather than signal.

When t-SNE produces a visualization of this character, it reflects the high-dimensional structure faithfully: the classes



(a) t-SNE, 330+ features (b) UMAP, 330+ features

Fig. 2: Dimensionality reduction visualization of the initial 330+ feature space. Orange points: minority *active-transacting* class. Blue: combined negative class. The absence of meaningful cluster structure—scattered distribution, extensive class overlap, diffuse boundaries, isolated outlier groups—indicates poor discriminative quality in the original feature space.



(a) t-SNE, 19 features (b) UMAP, 19 features

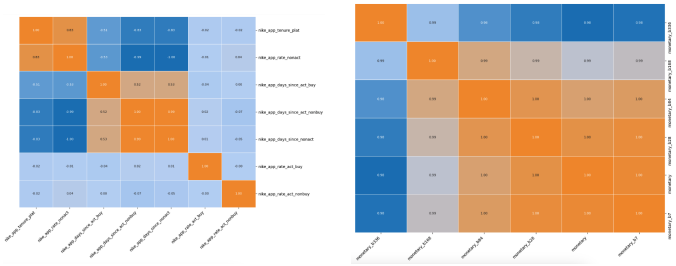
Fig. 3: Dimensionality reduction visualization following feature reduction to 19 features. The *active-transacting* class (orange) forms distinct spatial groupings absent from Figure 2. The characteristic snake-like manifold structure in the UMAP projection reflects temporal continuity in engagement features—a meaningful behavioral signal, not noise.

are not well-separated in $\mathbb{R}^d$ either. This explains why the RandomForestClassifier requires such deep, expensive trees to find any decision boundary at all.

Figure 3 shows the same projections following reduction to 19 features. The improvement is qualitatively clear: the positive class forms distinct spatial groupings, and overall cluster structure is more pronounced. The UMAP projection reveals characteristic snake-like curvilinear manifold structures. These are a well-understood consequence of sequential temporal data in nonlinear dimensionality reduction [5]: time-indexed engagement metrics naturally form one-dimensional manifolds (curves) as members trace smooth behavioral trajectories over time. Rather than indicating a problem, these structures confirm that temporal engagement features are capturing genuine behavioral dynamics. They also carry an important implication for algorithm selection: curved manifolds are challenging for axis-aligned tree splits but are naturally handled by neural network architectures [10].

B. Pearson Correlation Analysis

We compute pairwise Pearson correlations [11] within each platform feature group and across transactional and monetary feature families. The analysis reveals pervasive near-perfect



(a) Platform A feature group (b) Monetary feature group

Fig. 4: Pearson correlation heatmaps for representative feature groups. Orange: strong positive correlation. Blue: strong negative correlation. Near-perfect correlations ($|r| \geq 0.99$) are pervasive within groups, indicating high redundancy. Each such pair is a candidate for removal.

correlations, particularly among temporal window variants of the same base feature.

Figure 4 and Table I summarize the most extreme cases. A consistent structural pattern emerges: *days_since_inactive* and *rate_inactive* are perfectly anti-correlated ($r = -1.00$) within every platform group, reflecting the definitional relationship between recency and engagement rate—these are not two independent features but one feature expressed in two mathematically opposite forms. Similarly, temporal window variants of monetary, LTV, and transactional features are perfectly or near-perfectly correlated across all lookback windows, confirming that values accumulated over 7, 28, 84, 168, and 336 days encode largely redundant information. Contactability flags are perfectly correlated across all temporal windows ($r = 1.00$), indicating that membership contactability status does not meaningfully change across the observed time horizon.

C. Variance Filtering

We compute per-feature variance across the dataset and identify features with zero or near-zero variance. Table II lists the 14 such features identified in the numerical set. These are predominantly engagement *rate_active-transacting* and *rate_active-non-transacting* features across Platforms C and D, reflecting the fact that buying and non-buying engagement rates on these platforms are near-constant across the member population—providing no discriminative signal whatsoever while adding 14 dimensions of pure noise to the feature space.

D. Chi-Squared Statistical Testing

We apply chi-squared tests [12] to assess the statistical dependence between each discretized feature and the target label. Several features exhibit p-values of 1.0, indicating complete statistical independence from the target—most notably platform-specific rate and first-transaction order size features.

Conversely, LTV-related features exhibit the strongest associations, with chi-squared statistics exceeding 18,000 and $p \approx 0$. Monetary spend features ($\chi^2 = 17{,}713$) and total past demand features rank similarly highly. Platform tenure features show borderline statistical significance ($p \approx 0.04$).

TABLE I: Representative highly-correlated feature pairs across platforms and feature families. Correlations at $|r| \geq 0.99$ indicate near-complete linear redundancy; retaining both features adds model complexity without adding discriminative signal. Platform letters A–D represent four digital engagement platforms.

Feature 1	**Feature 2**	r
Platform engagement features		
plat_A_days_since_inactive	plat_A_rate_inactive	−1.00
plat_A_days_since_non-buy	plat_A_rate_inactive	−0.99
plat_C_days_since_non-buy	plat_C_days_since_inactive	1.00
plat_D_tenure	plat_D_rate_inactive	0.90
Contactability flags		
contactable	contactable_b7	1.00
contactable	contactable_b28	1.00
contactable	contactable_b84	1.00
contactable	contactable_b336	1.00
Monetary and LTV features		
monetary	monetary_b7	1.00
monetary_b7	monetary_b28	1.00
monetary_b28	monetary_b84	1.00
ltv	ltv_b7	0.99
ltv_b7	ltv_b28	0.98
Transaction features		
first_trans_units	first_trans_units_b7	1.00
first_trans_units	first_trans_units_b28	1.00
first_trans_units	first_trans_units_b84	1.00

These results confirm the centrality of lifetime economic value signals—LTV, monetary spend, past demand—as the primary discriminators of purchasing behavior, while flagging platform-specific rate features and certain first-transaction fields as candidates for removal.

VI. Results

A. Feature Space Comparison

The combined application of correlation filtering, variance thresholding, and chi-squared selection reduces the feature space from 330+ dimensions to 19 features—a reduction exceeding 95%. The retained features are dominated by behavioral recency signals (*days_since_**) and engagement rate features for the inactive state, augmented by LTV, monetary spend, and total past demand signals at a single representative temporal window.

The qualitative improvement in class separability is demonstrated in Figures 2 and 3: the positive class transitions from being indistinguishable from background noise to forming distinct spatial groupings. Importantly, this improvement is achieved by *removing* features, not adding them—confirming that the original feature space was harming rather than helping the classifier.

TABLE II: Features with zero or near-zero variance. Platform letters C and D represent two lower-engagement digital platforms. All features are engagement rate metrics for minority behavioral states—near-constant across the member population and therefore uninformative as classifiers.

Feature
plat_C_days_since_active_transacting
plat_C_rate_active_transacting
plat_D_days_since_active_transacting
plat_D_rate_active_transacting
plat_C_rate_active_non-transacting
plat_D_rate_active_non-transacting
plat_A_rate_active_transacting
plat_A_rate_active_non-transacting
plat_B_rate_active_non-transacting
plat_B_rate_active_transacting
plat_C_web_rate_active_transacting
plat_C_web_rate_active_non-transacting
plat_D_web_rate_active_transacting
plat_D_web_rate_active_non-transacting

TABLE III: Training time comparison across configurations. Feature reduction transforms a computationally intractable pipeline into one that completes in under 2 hours on significantly larger data. RF = RandomForestClassifier; NN = Neural Network (TensorFlow/GPU); DNF = Did Not Finish.

Configuration	Rows	Trials	Time
RF, 330+ features	300K	150	>18 hr
RF, 330+ features	31.7M	150	DNF (>16 hr)
RF, ∼120 feat., n_j=1	300K	5	∼60 min
RF, ∼120 feat., n_j=5	300K	5	∼15 min
NN (GPU), 333 features	1M	150	∼2 hr

B. Computational Efficiency

Table III summarizes the training time comparison across configurations. The results are stark. With the original 330+ feature space, 150 Optuna trials on 300,000 records required over 18 hours, and the same procedure on 31.7 million records failed to complete a single trial after 16 hours. Parallelization through the Optuna `n_jobs` parameter provided incremental improvement but did not address the fundamental bottleneck: each individual trial was too expensive due to the feature space complexity.

The GPU neural network result—150 trials in ∼2 hours on 1 million records—is noteworthy. This was achieved without the feature reduction described in this paper, suggesting that GPU acceleration and neural network architecture together provide an alternative path to tractability. The neural network result also underscores the feature space geometry finding: the curvilinear manifold structures observed in Figure 3 are precisely the type of structure that neural networks handle naturally, motivating further investigation of NN-based architectures for this problem.

C. Model Performance

With the reduced feature set and class weighting applied to address imbalance [9], the best RandomForestClassifier configuration achieves F1 scores of 0.98 for the *inactive* class and 0.85 for the *active-non-transacting* class. The *active-transacting* class remains challenging (F1 $\approx$ 0), motivating future investigation of downsampling strategies, focal loss, and neural network architectures with stronger inductive biases for minority class detection. A companion regression model predicting LTV achieves RMSE $\approx$ $1,627 at tree depth 20.

VII. Discussion

A. Feature Quantity vs. Feature Quality

A central finding of this work is that feature count alone does not determine model quality or computational tractability—feature *quality* does. A well-curated set of numerical engagement features is not inherently intractable for a RandomForestClassifier. What renders the original 330+ feature space computationally prohibitive is the preponderance of redundant, zero-variance, and statistically independent features that force the model to construct unnecessarily complex trees simply to locate the minority class through layers of noise. The 14 zero-variance features identified in Table II, for instance, contribute nothing but additional split candidates for every tree in every Optuna trial.

B. Temporal Window Redundancy as a Systematic Pattern

The correlation structure revealed in Table I reflects a systematic pattern common in retail engagement feature engineering: a base behavioral metric (e.g., days since last activity, total monetary spend) is replicated across multiple temporal lookback windows to capture both short- and long-term trends. In practice, for members whose behavior has been stable, these window variants are nearly identical—and their Pearson correlations approach 1.0 accordingly. This suggests a general design principle for temporal feature engineering: *compute window variants, but test for redundancy before including all variants in the model*. In our setting, retaining a single representative window per metric, guided by chi-squared significance, is sufficient to recover the predictive signal while eliminating the majority of the feature space.

C. Algorithm Selection Under Feature Space Geometry

The snake-like manifold structures visible in the UMAP projection (Figure 3) have a direct implication for algorithm selection. Sequential temporal engagement data naturally forms one-dimensional curves in the projected low-dimensional space: as engagement recency and rates evolve over time, members trace smooth continuous trajectories rather than occupying discrete clusters. Axis-aligned decision boundaries (RandomForest) are poorly suited to capture curved structure—any axis-aligned approximation of a curve requires many shallow cuts, increasing tree depth and training cost [3]. Neural networks, by contrast, compose non-linear transformations that can learn curved boundaries efficiently [10], [13]. This suggests that for the minority class detection task specifically, a neural network architecture may ultimately outperform RandomForest on the same reduced feature set.

D. Scalability and Infrastructure Cost

At production scale (30M+ records), the computational gap between a well-engineered and a poorly-engineered feature space is not a matter of hours but of feasibility. No amount of vertical or horizontal compute scaling compensates for a model that requires exponentially deep trees to find a signal that barely exists in the feature representation. In distributed computing environments where cluster costs accrue by core-hour, reducing per-trial training time from hours to minutes has direct and substantial financial implications at production scale. The feature engineering work described here represents a direct cost reduction in both engineering iteration time and cloud infrastructure expenditure.

E. Limitations

The present analysis uses stratified samples of up to 1 million records rather than the full production dataset, and full-scale validation on the complete 39.5M-record dataset remains future work. The 19-feature set was identified through visualization-guided analysis rather than formal ablation studies; the marginal contribution of each individual feature has not been precisely quantified. The *active-transacting* F1 score of approximately 0.0 indicates that feature engineering is necessary but not sufficient—further work on class imbalance strategy and model architecture is needed to fully solve the minority class detection problem. Finally, platform and feature identifiers have been anonymized throughout; readers seeking to apply this methodology should adapt the diagnostic workflow to their own platform-specific feature structures.

VIII. Conclusions

We have presented a systematic methodology for diagnosing and resolving feature space noise in a large-scale customer action propensity classification system. By combining t-SNE and UMAP visualization, Pearson correlation analysis, variance filtering, and chi-squared statistical testing, we reduce a 330+ dimensional noisy feature space to 19 high-signal features—achieving over 95% dimensionality reduction and transforming a computationally intractable training pipeline into one capable of completing 150 hyperparameter optimization trials in under 2 hours on 1 million training examples.

The core practical message of this work is: *when a production machine learning pipeline becomes computationally intractable, the correct first response is to examine the feature space, not to scale the infrastructure*. In high-dimensional, severely imbalanced settings—which are the norm rather than the exception in real-world LTV modeling—feature noise is frequently the binding constraint, and systematic feature space diagnosis is the highest-leverage intervention available.

Future work includes formal ablation studies quantifying per-feature marginal contributions, full-scale validation on the complete dataset, investigation of neural network architectures suited to the curvilinear manifold geometry identified in the reduced feature space, and comparison of minority class handling strategies including downsampling [9] and weighted loss functions.

Acknowledgments

The author thanks the data science and engineering team members whose collaborative work in data engineering, model development, and infrastructure debugging informed and motivated this research.

References

[1] S. Gupta, D. Hanssens, B. Hardie, W. Kahn, V. Kumar, N. Lin, N. Ravishanker, and S. Sriram, "Modeling customer lifetime value," *Journal of Service Research*, vol. 9, no. 2, pp. 139–155, 2006.

[2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.

[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[4] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache Spark: A unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[5] L. J. P. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[6] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.

[12] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in *Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 1995, pp. 388–391.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.