Customer Feedback Classification with Machine Learning

Video Presentation

Tianyi Tong — z5517006
Qingyang Zheng — z5490841
Junle Zhao — z5447039
Yuchen Du — z5409061
Zheyu Gu — z5546853

1 INTRODUCTION

Effective customer feedback management is the key to improve product quality and service responsiveness. Manual feedback classification is time-consuming and inefficient, especially when the amount of data across departments is large. Automating this process using machine learning speeds up response times and ensures that feedback is delivered to the right team. Our task developed a multi-class classification model to classify feedback into 28 product departments from 300-dimension text embeddings. Our primary challenges include class imbalance (unbalanced label distribution), evaluation (using weighted cross-entropy loss instead of accuracy), distribution shift (real-world test set introduces domain change), and feature importance (identifying which text features are informative for classification).

2 EDA and Literature Review

The data set used in this article has been checked to have nothing to clean. In this paper, KBest (chi2) and PCA are used for feature and importance analysis. The dimensions identified by Kbest are most relevant to the class labels, PCA reduces the data by 300 dimensions, and the top 100 components capture more than 90% of the variance. The simultaneous analysis in this paper reveals that the class imbalance is severe, with some categories only accounting for 1–2% of the feedback, while other categories dominate. About a third of the categories each account for less than 2% of the data. In the absence of intervention, this would bias the model towards the majority class. To alleviate this problem, we use techniques such as class weighting, SMOTE oversampling, and downsampling.

Imbalanced classification results in insufficient accuracy. Therefore we used Weighted Cross-Entropy Loss, Macro-F1 Score, Weighted-F1 Score. The weighted cross-entropy loss is employed during training for highlighting the minority class, related to other class-balanced approaches [1]. The Macro-F1 score is employed to evaluate the performance of every category and is not overshadowed by frequent categories. The weighted-F1 score gives weighted average performance, considering class proportions.

A preliminary test of set 2 reveals that its distribution is skewed relative to the training set. To alleviate this problem, we explore methods like robust verification, pseudo-labeling, and domain adaptation in order to sustain performance in real-world conditions.

2.1 Related Works

In this study, the MLP model, as one of several classification algorithms, is used to classify waste, demonstrating its effectiveness in complex pattern recognition tasks [2].

The softmax function is usually used in the output layer of the neural network to convert the output of the model into the probability of the occurrence of various diseases (such as pneumonia), so as to realize the multi-classification detection of X-ray images [3].

Support Vector Machine (SVM) achieves efficient pattern recognition in multivariate classification tasks by constructing an optimal hyperplane, and shows good classification performance especially in high-dimensional feature space [4]. For high-dimensional data classification, a variety of intelligent optimization methods are proposed to improve the classification effect of SVM, and alleviate the computational burden and classification performance degradation caused by high-dimensional features [5].

By efficiently processing categorical and continuous features in multivariate models, CatBoost shows its powerful modeling ability on complex and multi-dimensional datasets, which is especially suitable for classification and regression tasks in high-dimensional feature spaces [6]. CatBoost is proposed as a class-specific gradient boosting algorithm, which improves performance and stability in multivariate classification and regression tasks through innovative class-specific feature processing and an efficient training mechanism [7].

3 METHODOLOGY

3.1 Data Balancing and Feature Preprocessing

SMOTETomek vs. Oversampling. Some classes have fewer than 20 samples. For MLP/SVM, we often use SMOTETomek [8] or moderate oversampling, though for SVM too many synthetic samples may hurt the margin. For CatBoost, `auto_class_weights` plus simple oversampling for extremely rare classes suffices.

PCA. We reduce 300-dim features to 100 principal components (about 90% variance) for MLP. For SVM, we initially tested the same PCA approach but found that it did not improve Weighted CE and thus did not incorporate it in the final model. CatBoost typically does not benefit from PCA.

3.2 Multilayer Perceptron (MLP)

Principle. Each layer computes:

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}, \quad a^{(l)} = f\big(z^{(l)}\big).$$

We use ReLU as $f(\cdot)$. The final layer has 28 logits, passed through softmax.

Configuration. A 3-layer MLP (256-128-64) uses ReLU, dropout(0.5), L2 regularization, and Adam (lr $= 10^{-3}$). Up to 50 epochs, early stopping. SMOTETomek+PCA(100D) can be applied first.

3.3 Support Vector Machine (SVM)

Principle (OvR). For $K = 28$ classes, we train 28 binary SVMs, each solving:

$$\min_{w,b} \ \tfrac{1}{2}\|w\|^2 + C\sum \xi_i, \quad y_i^{(k)}(w^T x_i + b) \geq 1 - \xi_i.$$

Implementation. We adopt an RBF kernel and search (C, γ) in small grids. `probability=True` provides per-class probabilities for Weighted CE. Partial SMOTE or weighting addresses imbalance.

3.4 CatBoost

Key Idea. CatBoost iteratively builds M trees, each fitting the residual from previous iterations:

$$F_M(x) = \sum_{m=1}^{M} \eta\, h_m(x),$$

optimizing Weighted CE:

$$\mathcal{L} = -\tfrac{1}{N}\sum_{i=1}^{N} w_{y_i}\ \log p_{i,y_i}.$$

We set`auto_class_weights="Balanced"` and oversample classes with $<$ 25 samples to address severe label imbalance.

Hyperparameter Search & Final Selection

We performed a 5-fold cross-validation (CV) on the training data to minimize *Weighted CE.* The following hyperparameters were explored in a small grid:

$$\texttt{max\_depth} \in \{4, 6, 8\}, \quad \texttt{learning\_rate} \in \{0.03, 0.1\}.$$

$$\texttt{iterations} \in \{500, 1000\}, \quad \texttt{l2\_leaf\_reg} \in \{1, 3\}.$$

The final configuration provided a reasonable trade-off for training speed, minority-class recall, as well as overall Weighted CE. We also applied early stopping (patience=50) for preventing overfit. Practically, higher iteration counts or deeper trees yielded decreasing benefits.

3.5 Evaluation Metrics & Model Selection

We choose **Weighted CE** as our main metric which emphasizes errors on rare classes via higher weights. We also report **Accuracy** and **Macro-F1** for secondary insights. However, since accuracy can be misleading in skewed data, the final choice is driven by the model's *lowest Weighted CE* on the test set.

4 RESULTS

4.1 Baseline and Model Comparisons

We evaluate on the 202 labeled samples of Test 2. Table 1 combines Accuracy, F1-scores, and Weighted CE in one view, including two trivial baselines:

Table 1: Comparisons of Baselines (Random, Majority) vs. Our Models (MLP, SVM, CatBoost) on 202 labeled Test 2

Classifier	**Accuracy**	**Macro-F1**	**Weighted-F1**	**Weighted CE**
Random Baseline	0.0248	0.0162	0.0329	1.02 (approx)
Majority Baseline	0.0446	0.0036	0.0038	0.94 (approx)
MLP	0.67	0.56	0.68	0.40
SVM	0.65	0.42	0.65	0.22
CatBoost	0.64	0.53	0.65	**0.19**

While MLP attains the highest accuracy (0.67), it yields the worst Weighted CE (0.40). CatBoost is best at Weighted CE (0.19) which focuses more on minority classes. SVM is moderate (CE=0.22).

Test1 vs. Test2 (Unlabeled). We also compute Weighted CE on the unlabeled parts of Test 1 (1k) and Test 2 (1.8k) via internal checks; CatBoost consistently shows a small edge over MLP and SVM.

4.2 Pseudo-Labeling Observations

For CatBoost, we briefly tested a two-phase pseudo-labeling on unlabeled Test 2 data (thresholds=0.95, 0.88). However, *macro-F1 dropped* from roughly 0.54 to 0.45 after the second stage, which means that the label noise can degrade minority performance under distribution shift.

4.3 Brief Ensemble Exploration

While the CatBoost model was the best according to Weighted CE, we did try a simple ensemble approach. Namely, we combined the three trained classifiers (SVM, MLP, CatBoost) using *soft voting*, i.e., averaging predicted class probabilities prior to selecting the most probable label. Surprisingly enough, the ensemble did not have a positive effect on Weighted CE for either Test 2's labeled subset (202 points) or for our internal validation set. CE stayed around 0.19–0.20, and macro-F1 was almost indistinguishable from CatBoost alone. We propose that the decision boundaries were already captured effectively by CatBoost, and MLP/SVM additions provided conflicting predictions for the minority classes. More advanced ensembling or stacked generalization can be tried in the course of followup research, but for the sake of brevity, we went with the single CatBoost model.

4.4 Final Model Selection & Submission

As Weighted CE is our principal metric, we refer to Table 1 for the *202 labeled subset* of Test 2. CatBoost attains the lowest Weighted CE (0.19), outperforming MLP (0.40) and SVM (0.22). Although MLP yields higher Accuracy (0.67) and SVM has a moderate CE, CatBoost provides the best overall Weighted CE performance, which aligns with the project instructions emphasizing minority-class weighting.

Hence, we choose CatBoost as our final model for the final model.

5 DISCUSSION

5.1 Model Insights

MLP. Good overall accuracy but high Weighted CE indicates it struggles with rare classes. **SVM.** Moderate CE (0.22), overshadowed by CatBoost for minority recall. **CatBoost.** Lowest CE (0.19). Automatically reweights minor classes, but can be resource-intensive.

5.2 Distribution Shift and Pseudo-Labeling

Even though we had a meticulously crafted training set, real-world deployment often reveals *distribution shift* [9]—the data distribution at inference may be quite different from that during training. In Test 2, we saw that some feature dimensions drifted from their original ranges (covariate shift), and that the relative frequencies of minority classes changed significantly (label distribution shift). As a result, we propose that there is a *joint distribution shift*, because both feature values and label distributions have deviated which making it even harder for the model to generalize to new data.

To address these issues, we tried pseudo-labeling on the unlabeled portion of Test 2. First, we used CatBoost to label only the most confident samples, then retrained on the combined set (original + pseudo-labeled). Although this approach expanded coverage, we noticed *noise amplification* when the threshold was too low. Minority classes were especially prone to mislabeling, causing Weighted CE to rise slightly (from 0.19 to 0.21).

5.3 Limitations and Future Directions

Despite surpassing our baselines, the proposed approach has limitations. Class imbalance remains severe, and synthetic oversampling (e.g., SMOTE) may introduce unrealistic samples. Distribution shift further underscores the need for robust domain adaptation. Moving forward, we plan to integrate **focal loss** or **class-balanced loss** to emphasize minority classes. We will also explore more selective pseudo-labeling or adversarial domain adaptation (e.g., CORAL [10]) to mitigate label drift.

6 CONCLUSION

We tackled a 28-class feedback classification with severe imbalance and distribution shift, implementing MLP, SVM, and CatBoost. Our results show:

- All models surpass trivial baselines;
- CatBoost achieves the best Weighted CE (0.19), favoring minority classes;
- MLP has higher accuracy (0.67) but struggles more with minority classes (CE=0.40).

Overall, Weighted CE proves more appropriate than accuracy for skewed data. Future work includes domain adaptation, ensemble strategies, and improved pseudo-labeling.

Limitations and Future Work. We plan to refine thresholding for pseudo-labeling, explore focal/class-balanced losses, and implement domain adaptation as data distributions evolve.

References

[1] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9268–9277.

[2] N.-C. Yang and K.-L. Sung, “Non-intrusive load classification and recognition using soft-voting ensemble learning algorithm with decision tree, k-nearest neighbor algorithm and multilayer perceptron,” *IEEE Access*, vol. 11, pp. 94 506–94 520, 2023.

[3] A. Saraiva, D. Santos, N. Costa, J. Sousa, N. Ferreira, A. Valente, and S. Soares, “Models of learning to classify x-ray images for the detection of pneumonia using neural networks,” in *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies*, 2019.

[4] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[5] S. Ding and L. Chen, “Intelligent optimization methods for high-dimensional data classification for support vector machines,” *Intelligent Information Management*, vol. 2, no. 6, pp. 354–364, 2010.

[6] A. Rosales-Perez, S. Garcia, and F. Herrera, “Handling imbalanced classification problems with support vector machines via evolutionary bilevel optimization,” *IEEE Transactions on Cybernetics*, pp. 1–13, 2022.

[7] J. T. Hancock and T. M. Khoshgoftaar, “Catboost for big data: An interdisciplinary review,” *Journal of Big Data*, vol. 7, no. 1, 2020.

[8] 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.

[9] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, *Dataset Shift in Machine Learning*. MIT Press, 2009.

[10] B. Sun and K. Saenko, “Deep CORAL: Correlation alignment for deep domain adaptation,” in *European Conference on Computer Vision (ECCV) Workshops*, 2016, pp. 443–450.