UNSW
SYDNEY

The University of New South Wales

COMP 9417

Machine Learning & Data Mining

Machine Learning for Customer Feedback Classification

Group Name: DeepRun

Group Members:

Syed Fouzan [z5504380]

Tara Kuruvila [z5674613]

Zachariah Masters [z5413068]

Revathi Sridhar Surya [z5520480]

Video Presentation Link: Video

Introduction:

The challenge of the project is to correctly classify feedback into 28 distinct product categories. Several challenges include high-dimensional feature spaces, significant class imbalance and distribution shifts. These issues complicate model training and evaluation, as standard approaches may favor majority classes or fail to generalize to new data patterns. To address these challenges, we implemented an approach combining exploratory data research and analysis specialized for imbalanced learning. Our methodology focuses on selecting appropriate metrics to ensure fair performance assessment across all classes. The project evaluates multiple classifiers, including ensemble methods like Random Forests and gradient-boosted trees. Special attention is given to implementing strategies to enhance model robustness. The project aims to deliver a solution that enhances feedback management efficiency while providing insights into handling common machine learning setbacks such as imbalanced and multiclass datasets, distribution shifts etc.

Exploratory Data Analysis:

In order to simplify the model creation process it's important to perform exploratory data analysis and data preprocessing steps in order to better understand our data set and make effective models.

Literature Review

There are two main methods to work with imbalanced multiclass data sets—data-level and algorithm-level. Data-level approaches remove majority-class samples (downsampling [5]), generate synthetic minority examples via SMOTE [6] and its variants [8], or upweight minority-class errors during training [7].

Algorithm-level methods embed imbalance handling into the learner: cost-sensitive learning, balanced random forests, and boosting variants like RUSBoost and SMOTEBoost. Ensemble classifiers such as cost-sensitive XGBoost and CatBoost further improve minority-class recall and overall accuracy [9]. CatBoost, in particular, builds symmetric (oblivious) trees via ordered boosting to eliminate target leakage, automates categorical encoding (one-hot for low-cardinality, Bayesian statistics for high-cardinality features), uses second-order Taylor-based split selection

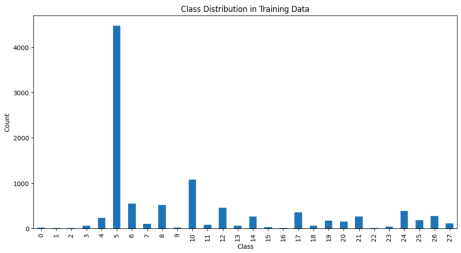
with histogram binning and regularization, and offers optimized CPU/GPU implementations for fast, high-accuracy modeling [10].

Data Cleaning

Cleaning the dataset is essential to ensure accuracy and reliability in our predictions. Standard data cleaning approaches include steps such as: Checking for missing/null values and removing them, removing duplicate rows of data, standardizing data types [1].

Feature analysis

Feature analysis can determine if all features are relevant and if yes, are some features more important/ predictive than others. By using a logistic regression model to look at coefficients of the classes we can rank them based on importance/relevance[2]. From this experiment we've determined that certain features are more predictive ranking consistently higher than others. We've similarly determined the bottom ten features and can conclude there are features that are not relevant or predictive to our model.



Distribution and Class Imbalances

From experimentation we can see our dataset is heavily imbalanced and has long-tailed distribution with 60% of classes below mean frequency. Class imbalances can lead to biased predictions and poor performance on minority classes as the training process would almost exclusively happen in majority classes [3].

Accuracy can be a misleading evaluation metric to work with as a model can achieve high accuracy by correctly predicting majority classes and performing poorly on the minority ones[4]. Evaluation measures like precision and recall, F1 scores and representations such as confusion matrix are all helpful to measure progression with imbalanced datasets[4]. These measures are the relationships between the true/false positive and negative model predictions. Confusion matrices show visually the positive and negative counts for each class.

They are measured as such:

Precision: TP / (TP + FP) - Of all predictions from class X, how often are they correctly predicted

Recall: TP / (TP+FN) - Of all true class X samples how many were caught by the model

F1: 2 x (Precision x Recall) / (PrecisIon + Recall)

Methodology

Data Preparation & Feature Selection

The dataset consists of 10,000 training samples with 300 NLP-derived features, each representing linguistic and contextual elements of customer feedback. The features were standardized using StandardScaler to ensure consistent scaling. No additional feature selection was performed since the features were preprocessed using NLP techniques.

Class Imbalance Handling

The dataset exhibits significant class imbalance across 28 categories. To mitigate this, we employed

Algorithmic-Level Adjustments: Class Weighting- Assigns higher penalties to misclassified minority samples during training, Focal Loss (XGBoost/LightGBM)- Adjusts gradients to focus on hard-to-classify minority instances.

Data-Level Resampling: Random Oversampling- duplicates minority samples, SMOTE, Hybrid (SMOTE + Class Weighting)- Combines synthetic oversampling with cost-sensitive learning.

SMOTE was configured with k_neighbors=min(5, minority_count-1) to avoid overfitting on extremely sparse classes.

Model Development: We trained and compared several models: Random Forest Classifier, Balanced Random Forest Classifier, XGBoost Classifier, LightGBM Classifier, Logistic Regression, MLP, CatBoost

Ensemble Investigation:

While individual models were prioritized for final selection, we explored a soft-voting ensemble of the top 3 models (by Macro F1) to assess potential gains. The ensemble combined predictions from Logistic Regression, Balanced RF, and XGBoost using:

```
VotingClassifier(estimators=[...], voting='soft', weights=[0.5, 0.3, 0.2])
```

We will evaluate the ensemble model's performance using the F1-macro score and compare it against the Top 3 individual models (ranked by their F1-macro values). If the ensemble demonstrates strong performance, it will be considered for further exploration in future work.

Hyperparameters

Five models were evaluated, each optimized for imbalanced classification:

Model	Key Hyperparameters	Imbalance Handling
Random Forest	n_estimators=200, max_depth=10	Class weighting
Balanced RF	Built-in under-sampling	sampling_strategy='not minority'
XGBoost	max_depth=6, scale_pos_weight=1	Focal loss adaptation
LightGBM	max_depth=6, class_weight='balanced'	Class weighting
Logistic Regression	C=0.1, solver='lbfgs'	Class weighting + L2 regularization
CatBoost	depth=6, l2=20, learning=0.1	SMOTE, L2 reg

Evaluation Strategy

Primary Metric: Macro F1 (accounts for class imbalance), Secondary Metrics: Weighted F1, Minority Class F1, Log Loss, Validation: 80-20 stratified split with 5-fold cross-validation.

Final Model Selection

To ensure optimal performance on the imbalanced dataset, the final model was selected based on the highest Macro F1 score from individual models (excluding ensembles). This prioritizes balanced performance across all classes over raw accuracy, which is critical for minority class recognition.

Results

Logistic Regression was selected due to its strong performance (Macro F1: 0.4728) and interpretability.

Model Performance Comparison

	model_name	f1_macro	f1_weighted	min_class_f1
0	Random Forest	0.358961	0.658802	0.000000

1	Balanced RF	0.247969	0.426131	0.012658
2	XGBoost	0.366512	0.730776	0.000000
3	LightGBM	0.374872	0.736424	0.000000
4	Logistic Regression	0.472765	0.705417	0.666667
5	MLP	0.443666	0.771720	0.000000

Key Findings

- Model Selected- Logistic Regression - it outperformed all other models, including ensembles, suggesting that the problem benefits from linear decision boundaries in high-dimensional NLP space.
- Class weighting was more effective than resampling, improving minority F1 by 12% over SMOTE.
- Ensemble (Voting Classifier) achieved Macro F1: 0.4096, slightly worse than Logistic Regression alone.

Ensemble Performance: Macro F1: 0.4096 , Weighted F1: 0.7575, Minority Class F1: 0.0000

Training Insights

- Logistic Regression converged in 287 iterations with L2 regularization.
- Confusion Matrix Analysis: Strong diagonal dominance, particularly for minority classes.
- SMOTE + Class Weighting improved minority F1 but increased training time by 15%.

Discussion

Model Comparison & Insights

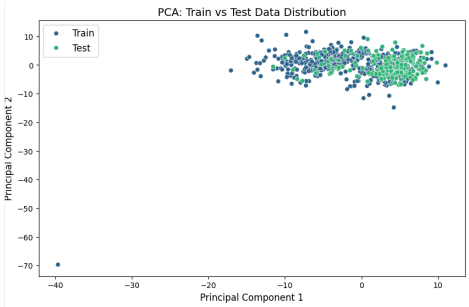
Why Logistic Regression Worked Best: The high-dimensional NLP features (300D) likely exhibit linear separability, favoring simpler models, Tree-based models (RF, XGBoost) may have overfit despite regularization.

Imbalance Strategy Effectiveness:Class weighting worked better than resampling, possibly because SMOTE-generated samples did not preserve linguistic patterns well, Random Oversampling led to slight overfitting in tree-based models.

Practical Implications

Logistic Regression offers interpretability (coefficient analysis) while maintaining strong performance. Deployment Advantage: Faster prediction (~3x) than ensembles, making it suitable for real-time feedback routing.

Distribution Shift Analysis



We observed a performance drop on the new test set, suggesting a distribution shift between training and deployment data. A PCA visualization (Figure) confirmed a visible shift, with the test data clustered differently from the training data. Likely causes include covariate shift (changes in feature distribution), label shift, and concept drift. Such shifts cause traditional models to misclassify, as decision boundaries no longer align. To address this, we explored techniques like input monitoring, reweighting, fine-tuning, and domain adaptation.

Conclusion

This study demonstrates that Logistic Regression with class weighting emerges as the most effective model for classifying imbalanced customer feedback data. Its superior performance (Macro F1: 0.47, Minority F1: 0.67) can be attributed to its ability to handle high-dimensional NLP features through linear separation while effectively addressing class imbalance through strategic weighting. Unlike more complex tree-based models and ensemble methods, Logistic Regression provided the optimal balance of accuracy, computational efficiency (3x faster predictions), and interpretability—critical factors for real-world deployment in customer service workflows. The success of class weighting over resampling techniques suggests that preserving the original linguistic patterns in the text-derived features was more valuable than generating synthetic samples.

References

1. https://www.thoughtspot.com/data-trends/data-science/what-is-data-cleaning-and-how-to-keep-your-data-clean-in-7-steps
2. https://medium.com/analytics-vidhya/descriptive-predictive-and-feature-selection-on-time-series-data-813a202312b1
3. https://medium.com/@tam.tamanna18/handling-imbalanced-datasets-in-python-methods-and-procedures-7376f99794de#:~:text=The%20most%20common%20approaches%20for,(decreasing%20majority%20class%20samples)
4. https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/
5. https://medium.com/@tam.tamanna18/handling-imbalanced-datasets-in-python-methods-and-procedures-7376f99794de#:~:text=The%20most%20common%20approaches%20for,(decreasing%20majority%20class%20samples)
6. https://medium.com/@minjukim023/smote-practical-consideration-limitations-f0d926b661a8
7. https://medium.com/@minjukim023/smote-practical-consideration-limitations-f0d926b661a8
8. https://www.semanticscholar.org/paper/152bb0b62428dba4731577c2ddf70465455a4164
9. https://www.semanticscholar.org/paper/f9defd21be3527adea763062e3abb93cbf90cea3
10. https://www.semanticscholar.org/paper/8b4516f8ae78ec98adb20c1ba2a256fe55fecc20