COMP9417 Project Report

Robust Feedback Classification under Imbalanced and Shifted Distributions

Group Name: 9417upup

Student Haoyue Bu (z5400609)

Student Xue Zhang (z5485499)

Student Jiacheng Chen (z5381497)

Student Minghao Bai(z5453461)

Student Muchen Wei(z5437335)

April 27, 2025

For the demonstration video, please refer to 9417 project video link.

1 Introduction

This project aims to develop a robust machine learning model that can automatically classify customer feedback into 28 product-related departments, ensuring that each comment can be accurately assigned to the corresponding department for processing. The main challenges faced by this project include serious class imbalance, a distribution shift between training and test sets, and the need to optimize classification performance between large sample classes and small sample classes. To address these challenges, the paper applies feature selection, random sampling techniques, Focal Loss indicators, advanced loss function DNN design, and comparison and optimization of three classification models, namely LR model, SVM model, and LowBais DNN model. Finally, by comparing the cross entropy indicator, the project finally chose the LowBais DNN model to achieve the goal. This report will show in detail the data exploration and analysis, a relevant review of the literature, the selection of the modeling method and its rationality, the experimental results of each model, and a comprehensive performance comparison, and finally give an overall summary and suggestions for future improvement directions.

2 Exploratory Data Analysis and Literature Review

2.1 Class Distribution Analysis

The paper counted the number of samples in each category and found that there was a serious imbalance between the categories. Some categories (such as category 5) had thousands of samples, while other categories such as category 1 and category 2 had only single-digit samples.

2.2 Feature value statistics and outlier detection

The paper showed the feature mean is close to 0 and the standard deviation is close to 1, indicating that the original features have been standardized, which is conducive to the stability of subsequent model training. Most of the maximum and minimum values of the features are distributed within ± 5, but there are also a few extreme values. Then, the paper counted the proportion of outliers under different thresholds. Overall, the proportion of outliers is very small (only about 0.006 within the range of ± 3), which will not affect model training on a large scale. Some features have slightly more outliers, which can be used as a reference for feature selection.

2.3 Redundant feature detection

In order to analyze the contribution of features to the classification task, the variance of each feature in the training set was first detected. The variance of all detected features was greater than 0.01, indicating that all features have good discrimination and no useless noise from the model. The top ten low-variance features may contribute very little to the model. In addition, the feature correlation matrix was analyzed, and the results showed that the correlation coefficient between any feature pair did not exceed 0.95, indicating that there were no obvious redundant feature pairs, and there would be no multicollinearity affecting logistic regression, linear models, etc. .

2.4 Analysis of importance of features

The importance scores of all features were calculated using random forests and the top 20 important features were visualized. This section helps understand which dimensions are most helpful for the model to distinguish categories and guide subsequent feature selection schemes.

2.5 PCA dimensionality reduction visualization

Using PCA to reduce high-dimensional features to two dimensions, it is found that although there is overlap between most categories of the data, some boundaries are still separable, indicating that further nonlinear modeling or feature engineering may be effective.

2.6 The impact of imbalanced data on model evaluation metrics and methods

Due to data imbalance, using only accuracy is just a simple statistical prediction of the proportion of correct samples without distinguishing different categories, which will mask the errors of minority categories. Therefore, these indicators are more appropriate, such as Macro-F1 (Pedregosa et al., 2011), Cross Entropy Loss (Goodfellow et al., 2016), Focal Loss (Lin et al., 2017). Existing studies about mitigation methods have proposed data-level methods, such as SMOTE (Kotsiantis et al., 2006) and ADASYN (Ramyachitra and Manikandan, 2015), which modify a few samples in the training data through undersampling, oversampling or hybrid methods to improve the category distribution; however, in the case of small sample size or high-dimensional features, it may cause noise problems (Gandomi and Yang, 2025). Recent studies have also adopted DNN-based methods to solve the category imbalance problem of high-dimensional data (Gandomi and Yang, 2025).

3 Methodology

3.1 Data Processing and Feature Selection

In this experiment, we first load the training dataset `X_train.csv` and the label `y_train.csv`, and then use the shuffle function to randomly shuffle the data order to avoid the influence of the order during training. Then, we use the `RandomForestClassifier` algorithm to train all features and calculate the importance of each feature. By sorting the feature importance, the top 200 most important features are selected. The initial experiment used 150 features, and the final experiment selected 280 features. After multiple experimental verifications, the selection of 200 features has the best effect, so it was finally decided to use 200 features as the input for model training. By reducing unnecessary features, the training efficiency and prediction accuracy of the model are improved. Then using `StandardScaler`standardized the selected features. Finally, the data is allocated to the training set at a ratio of 80% and the validation set at a ratio of 20% by `train_test_split`.

3.2 Base Models

3.2.1 SVM Base Model

The base model of this approach is a Support Vector Machine (SVM) classifier, using the RBF kernel (Radial Basis Function) to handle the data. The model's hyperparameters include $C = 1$ and $\gamma =$ 'scale', where C controls the model's fit to the training data; a larger C makes the model focus more on the training set, which may lead to overfitting. γ controls the influence of data points in the high-dimensional space, and using 'scale' adjusts it automatically based on the number of features. To address class imbalance, the model sets `class_weight='balanced'`, which automatically adjusts the weight of different classes. `StandardScaler` is used for standardization to ensure that the feature data has a mean of 0 and a standard deviation of 1, helping the model train more stably. However, the results of using only SVM for imbalanced data classification are not good. training set accuracy is 0.70, F1 score is 0.4584, and most of the minor classes have no correct predictions.

3.2.2 Logistic Regression Base Model

The base model of this approach is a simple logistic regression model, which uses StandardScaler for feature scaling to ensure that each feature has a mean of 0 and a standard deviation of 1, thereby improving the model's training stability and convergence speed. The model's hyperparameter settings include `max_iter=1000`, ensuring that the model can iterate sufficiently on complex datasets to avoid failing to converge with the default iteration limit. `random_state=42` is used to ensure consistent results when splitting the dataset into training and validation sets. The model is trained on the standardized data, and predictions are made using the validation set. Model evaluation uses a confusion matrix and classification report, providing accuracy, re-

call, F1 score, and other metrics to help analyze the model's performance across different classes. The final result yields a validation accuracy of 0.75 and an F1 score of 0.4109, with poor performance on small class categories.

3.2.3 Baseline DNN Model

The base DNN model used in the study is a fully connected neural network with two hidden layers with 512 neurons each and ReLU activation function. Therefore, it utilizes the Cross-Entropy Loss as the loss function and the Adam optimizer with the learning rate set at 3×10^{-4}. The model is trained for 100 epochs with 32 batch size. The data was preprocessed by using the `StandardScaler` and then it is split into 80% training data and 20 % of the data was used for validation. It is a basic and efficient reference model that achieves fast training and gives reasonable accuracy on most of the classes which offer further enhancements.

3.3 Shift Detect Analysis

In Generally, traditional machine learning models assume that training data and test data are generated independently from the same distribution(i.i.d. assumption). The decision boundary learned by the model is optimized for the training distribution and may not be well generalized to the new distribution, reducing the accuracy of the model, especially on minority classes. The displacement detection analysis performed by the DNN model showed significant differences in label distribution between the training and test sets. For instance:

- Class 12: 0.0457 (train) vs. 0.2129 (test)
- Class 25: 0.0184 (train) vs. 0.1287 (test)
- Class 5: 0.4479 (train) vs. 0.0446 (test)

These differences suggest the problem of label shift in which the training and test distributions are different, thus limiting generalization.

To tackle this, SMOTE was employed to oversample the minority classes, while penalty-based loss functions were applied to enhance the classification of the same.

Additionally, a DNN with Focal Loss and RF-based feature selection was designed to improve the performance against label shift. This methodology constitutes the core innovation of the proposed approach and is elaborated in the following sections.

Some of the strategies discussed to enhance the accuracy of the models and address the problems of class imbalance and distribution shift include:

Data Oversampling:

- **Random Oversampling:** This technique can ensure that the number of samples in the minority classes was increased to match that of the majority classes. However, this approach has a disadvantage: models such as Logistic Regression may overfit because the same data are repeated with no new information added.
- **Synthetic Minority Over-sampling Technique(** It involved synthesizing new samples by creating new instances from the existing ones of the minority class. While SMOTE works well in increasing the diversity of the data, if the original dataset contains noise or has low density may reduce model performance.

As mentioned above, oversampling was applied selectively based on the characteristics of the models.

Loss Function Enhancement: Focal Loss is the method increases the focus on the minority classes by down-weighting the influence of samples that are easy to classify during training through the use of modulating factors. These strategies were incorporated into different model pipelines depending on the complexity of the models and their sensitivity to imbalance, and all models were tested using the validation and test sets.

3.4 Improved Models

3.4.1 Improved SVM Model

The improved SVM model selects only 200 most important features based on Random Forest feature importance which helps in addressing the high dimensionality problem during the training phase. In an effort to overcome class imbalance problem, Random Oversampling and SMOTE is applied. Random Oversampling repeats identical copies of the minority class sample while SMOTE creates new sample cases to make the distribution of classes better and overcome the risk of overfitting. These enhancements are accumulated to enhance the model's generalization for all the classes but in particular for the minority classes.

3.4.2 Improved Logistic Regression Model

The improved logistic regression model introduces a feature selection process to reduce the dimension and minimize the impact of irrelevant or redundant features, thereby improving computational efficiency and reducing the risk of overfitting. Although SMOTE and random oversampling were originally used to solve the class imbalance problem, they led to performance degradation due to overfitting and noise generation, with an accuracy of 0.64 for RandomOverSampler and 0.65 for smote. Therefore, only feature selection is retained as an enhancement strategy for the logistic regression model.

3.4.3 Improved Deep Neural Network with Focal Loss

The model adopts a DNN with four fully connected layers. The input consists of 200 features, followed by three hidden layers of 512 neurons each (`hidden_dim=512`), using ReLU activation and Dropout to mitigate vanishing gradients and overfitting. The output layer classifies into 28 categories. The optimizer is Adam with a learning rate of `3e-5`, and learning rate adjustments are handled by a `ReduceLROnPlateau` scheduler. The model is trained with a batch size of 32 over 100 epochs.

Focal Loss is chosen over traditional Cross Entropy Loss because the dataset suffers from class imbalance, and standard cross-entropy loss tends to cause the model to become biased toward larger classes while ignoring smaller ones. Focal Loss mitigates this issue by improving the recognition of small class samples, focusing more on difficult-to-classify examples. This loss function contains two important hyperparameters: 'alpha' and 'gamma'. The 'alpha' parameter serves as a class balance factor, typically set to a small value to reduce the influence of larger classes. The 'gamma' parameter adjusts the attention given to difficult samples—larger values of 'gamma' increase focus on harder-to-classify examples, guiding the model to concentrate on error-prone instances during training, which in turn improves classification accuracy. Focal Loss is particularly well-suited for situations where the dataset has highly imbalanced categories. During the training process, the values of 'alpha' and 'gamma' were iteratively adjusted, and the model ultimately selected 'alpha=0.25' and 'gamma=2'. The choice of these hyperparameters was based on multiple experiments and careful evaluation of the model's performance on the validation set.

4 Discussion

4.1 Base Models Results

4.1.1 SVM Base Model

The initial SVM model achieved a validation accuracy of 0.70 and with a weighted F1 score of 0.7391 and an Unweighted F1 score of 0.4109. However, many small classes receiving near-zero precision and recall, indicating that the model was heavily biased toward majority classes.

4.1.2 Logistic Regression Base Model

The initial Logistic Regression model achieved a validation accuracy of 0.75 and an F1 score of 0.4109. The

performance in minority classes remained unsatisfactory, as reflected by very low recall and F1 scores for rare classes.

4.1.3 DNN Base Model

On the validation set, the DNN model had an overall accuracy of 0.78, the weighted F1 score of 0.7706, and the unweighted F1 score of 0.4395. A close value of the F1 weighted score to the overall accuracy also confirms that the model is good at the majority classes. However, the unweighted F1 score, which is lower than the accuracy, indicates that the model can have issues with the minority classes(as shown in Figure 2). Class 5, 6, 10 and class 12 performed well with F1 scores greater than 0.75. However, the minority classes 0, 1, 2, 9, 16, and 22 had precision, recall, and F1 scores near to zero because the model did not identify any samples from these classes correctly. Therefore, while the performance of the basic DNN model is good on the frequently occurring types, the result for low-probability items is significantly worse, which is why it is necessary to apply additional measures for increasing their accuracy.

4.2 Improved Models Results

4.2.1 Over-sample SVM Model Results

The improved SVM model achieved an accuracy of 0.77, with an unweighted F1 of 0.4618. This indicates strong performance on frequent classes, though rare classes remain difficult to predict.

From the Confusion Matrix (Figure 1), we observe improvements for small classes, especially class 5, which now has higher precision and recall. More correct predictions along the diagonal suggest better classification overall. However, the performance gap between large and small classes remains, as seen from the difference between weighted and unweighted F1 scores.

4.2.2 Improved Logistic Regression Results

The improved Logistic Regression model achieved 0.77 accuracy, with weighted F1 at 0.7568 and unweighted F1 at 0.4657. This reflects good overall performance but reveals issues with class imbalance.

From the Confusion Matrix (Figure 3), we see that class 1 reached perfect scores (1.00 precision/recall), while class 0 had 0.00 for both. Frequent classes such as class 5 and 6 performed well (precision/recall both $>$ 0.75). Most misclassifications occur in minority classes.

4.2.3 Low Bias DNN Results

The Low Bias DNN achieved 0.79 accuracy, weighted F1 of 0.7745, and unweighted F1 of 0.451. While it handles frequent classes well, small classes like 0, 1, 9, and 13 show 0 precision and recall.

From the Classification Report (Figure 5), we observe that class 5 achieved strong results (92% precision, 96% recall, 0.94 F1), and class 10 also performed well (78% precision, 79% recall, 0.79 F1). Overall, macro F1 is 0.47, while weighted F1 is 0.77, highlighting the model's bias toward large classes.

The introduction of Focal Loss enabled correct classification for some small classes. Compared to baseline methods, it improved recall and F1 scores in a few rare categories.

4.3 Models Comparison

4.3.1 Test Result

The test results of the three models on the first 202 rows of `X_test_2.csv` are as follows:

Model	Test Accuracy	Test unweighted F1 Score	Test Cross-Entropy Loss
Improved LR	0.58	0.29	0.0298
Oversample SVM	0.64	0.43	0.0204
Low Bias DNN	0.64	0.46	0.0203

Table 1: Test set performance on the first 202 samples of `X_test_2.csv`.

By comparing the test results, the model with the best performance in the test set is the Low Bias DNN model, as shown in the graphs of the results in Figure 6 and Figure 7.

4.3.2 Validation Dataset Result

The validation results of the three models on the 20% hold-out validation set are summarized below.

Model	Validation Accuracy	Validation unweighted F1 Score	Validation Cross-Entropy Loss
Improved LR	0.77	0.4657	0.0115
Oversample SVM	0.77	0.4618	0.0070
Low Bias DNN	0.78	0.4726	0.0068

Table 2: Validation set performance of different models.

Tables show that the low-bias DNN model has the highest F1 score (weighted and unweighted), indicating that it is better at classifying the minority class. In addition, the cross-entropy loss value of the DNN model has the lowest error value on both the validation set (0.0068) and the test set (0.0203) compared to the other two models, which means that the probability of incorrect prediction is the lowest.

4.3.3 Model Comparison Discussion

On the validation set, the low-bias DNN model has the lowest weighted logarithmic loss (0.0068), outperforming the over-sampled SVM (0.0070) and the improved logistic regression (0.0115), showing higher probability output confidence and classification stability. On the test set, the DNN model also has the lowest weighted logarithmic loss (0.0203), slightly outperforming the SVM (0.0204) and significantly outperforming the logistic regression (0.0298), indicating that it can better maintain performance when deployed in practice. Although the overall accuracy of the SVM and DNN on the test set is similar (both 0.64), the DNN has a higher macro F1 score (0.46 vs 0.43) and lower cross entropy, demonstrating that the DNN has a comprehensive advantage in minority category recognition and prediction confidence. In addition, the logistic regression model is significantly inferior to the DNN and SVM in cross entropy, indicating that its output probability distribution is poor under multi-category imbalanced data, resulting in prediction errors and low confidence. thereforethe low-bias DNN model showed better balance and generalization ability on both the training and test set, and was the ultimate optimal choice.

5 Conclusion

Although the basic SVM and LR models achieved a moderate level of overall accuracy, they did not perform well when dealing with a small number of categories, with a low Unweighted F1, exposing the serious impact of category imbalance. By introducing data enhancement techniques such as Random Over Sampler and SMOTE, the performance difference between large and small sample categories was alleviated, especially the weighted F1 score of the SVM model was improved. However, there are still some difficulties in identifying extremely rare categories. Combined with random forest feature selection, redundant features are effectively reduced, training efficiency is improved, and overfitting problems are alleviated.

Finally, the Focal Loss-based DNN model we proposed achieved an accuracy of 0.79 and a Macro F1 score of 0.45 on the validation set, and the Weighted Log Loss was only 0.0068, showing extremely high prediction confidence and balance. In contrast, the improved SVM model has a Weighted Log Loss of 0.007 on the validation set, and the logistic regression model has a Weighted Log Loss of 0.0115, both higher than the DNN model, indicating that DNN is also superior in the quality of probability output.

On the test set, the DNN model still performs best, with Macro F1 reaching 0.46 and Weighted Log Loss of 0.0203, which is also better than SVM (0.0204) and logistic regression (0.0298). This shows that DNN not only performs well in the training phase, but also maintains good generalization ability in the case of distribution shift. Thereforethe DNN model provides a foundation for the subsequent development of more complex feedback classification systems. If having more time, the project will do Ensemble Learning and combine existing models for weighted fusion to improve overall stability and minority category recognition capabilities.

References

[1] A. Gandomi, X.-S. Yang, "Black swan events and handling imbalanced data," referenced in *Balanced Datasets under Seismic Hazards*, 2025.

[2] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[3] Batista, G. E. A. P. A., Prati, R. C., Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 6(1), 20-29.

[4] Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

[5] Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215-242.

[6] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[7] D. Ramyachitra and P. Manikandan, "Imbalanced dataset classification and solutions: A review," *International Journal of Computer Applications*, vol. 116, no. 20, pp. 11–15, 2015.

[8] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.

[9] He, H., Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284.

[10] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal Loss for Dense Object Detection. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2980–2988.

[11] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

[12] Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision (ICCV), 2980-2988.

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[14] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, no. 1, pp. 25–36, 2006.

Appendix: All Figures

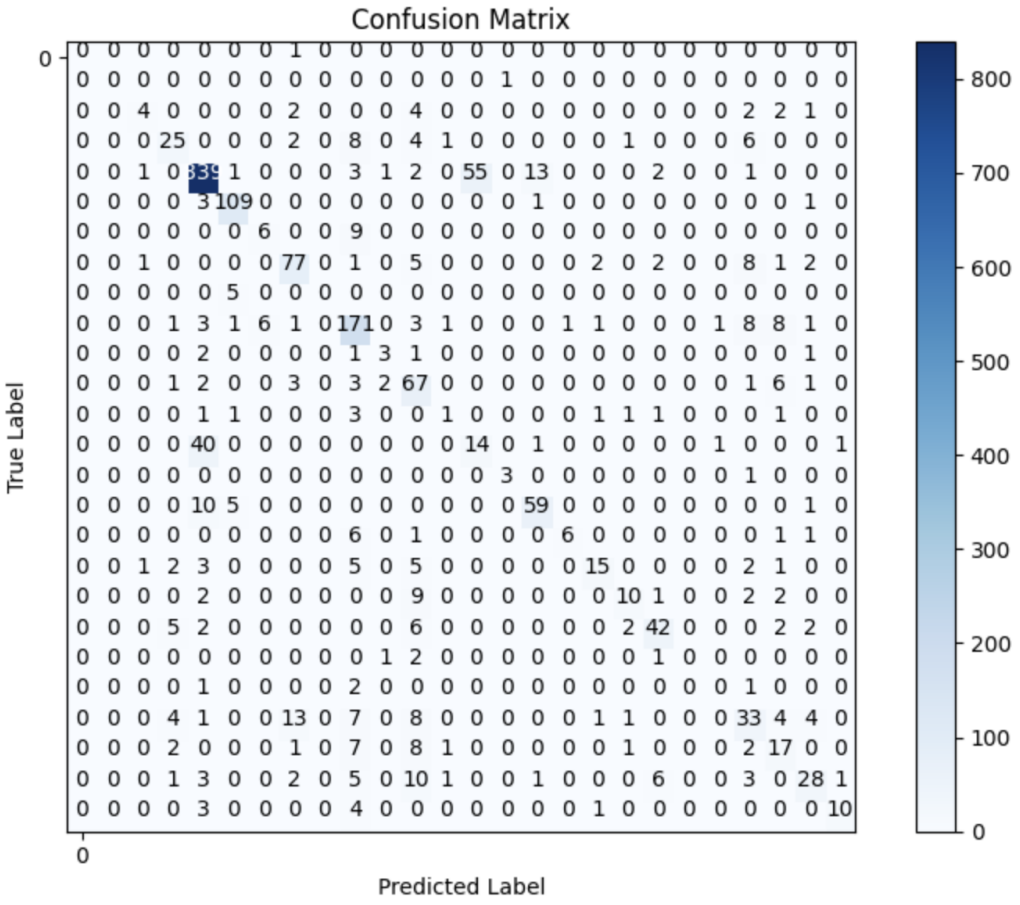

Figure 1: Confusion Matrix for Over-sample SVM Model

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	3
1	0.00	0.00	0.00	1
2	0.00	0.00	0.00	1
3	0.38	0.38	0.38	16
4	0.48	0.47	0.47	49
5	0.91	0.95	0.93	908
6	0.88	0.96	0.92	112
7	0.60	0.53	0.56	17
8	0.77	0.74	0.76	107
9	0.00	0.00	0.00	4
10	0.76	0.81	0.78	203
11	0.36	0.71	0.48	7
12	0.64	0.64	0.64	102
13	0.14	0.08	0.10	13
14	0.12	0.06	0.08	53
15	0.50	0.25	0.33	4
16	0.00	0.00	0.00	2
17	0.76	0.73	0.74	70
18	0.71	0.62	0.67	8
19	0.57	0.47	0.52	34
20	0.48	0.52	0.50	27
21	0.75	0.83	0.78	46
22	0.00	0.00	0.00	1
23	0.33	0.50	0.40	6
24	0.49	0.49	0.49	81
25	0.44	0.35	0.39	40
26	0.66	0.62	0.64	63
27	0.83	0.68	0.75	22
accuracy			0.78	2000
macro avg	0.45	0.44	0.44	2000
weighted avg	0.76	0.78	0.77	2000

Weighted F1 Score: 0.7706

Figure 2: Classification Report for Base DNN Model

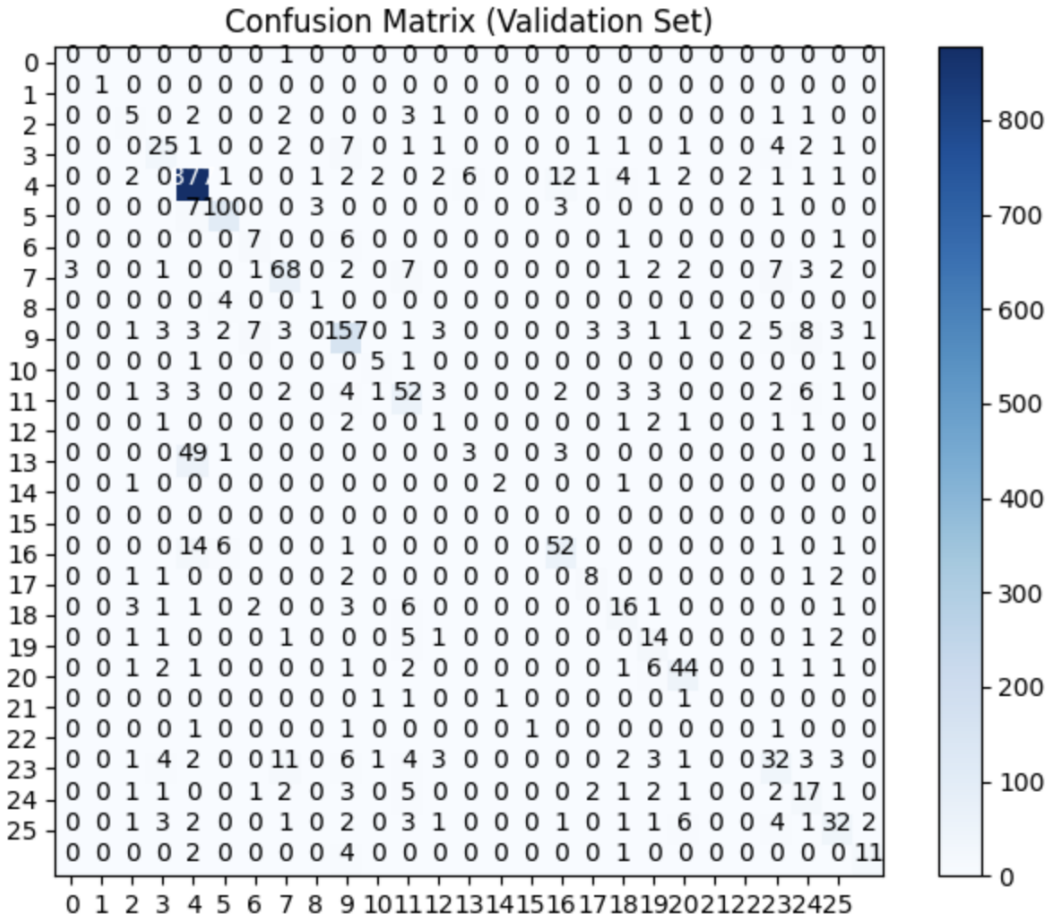

Figure 3: Confusion Matrix for Improved Logistic Regression Model

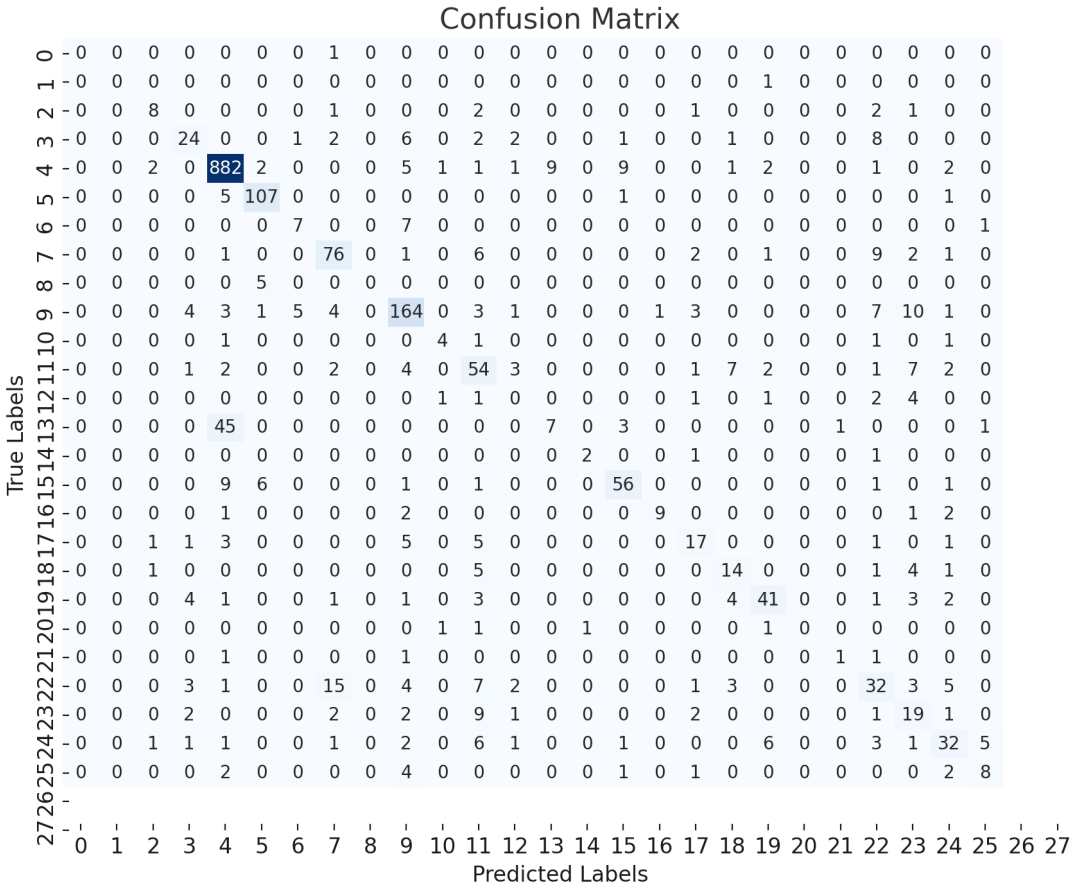

Figure 4: Confusion Matrix for Low Bias DNN

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	0.00	0.00	0.00	1
3	0.62	0.53	0.57	15
4	0.60	0.51	0.55	47
5	0.92	0.96	0.94	918
6	0.88	0.94	0.91	114
7	0.54	0.47	0.50	15
8	0.72	0.77	0.75	99
9	0.00	0.00	0.00	5
10	0.78	0.79	0.79	207
11	0.57	0.50	0.53	8
12	0.50	0.63	0.56	86
13	0.00	0.00	0.00	10
14	0.44	0.12	0.19	57
15	0.67	0.50	0.57	4
17	0.78	0.75	0.76	75
18	0.90	0.60	0.72	15
19	0.57	0.50	0.53	34
20	0.47	0.54	0.50	26
21	0.75	0.67	0.71	61
22	0.00	0.00	0.00	4
23	0.50	0.25	0.33	4
24	0.44	0.42	0.43	76
25	0.35	0.49	0.40	39
26	0.58	0.52	0.55	61
27	0.53	0.44	0.48	18
accuracy			0.78	2000
macro avg	0.50	0.46	0.47	2000
weighted avg	0.77	0.78	0.77	2000

F1 Score (Weighted): 0.7729
F1 Score (Unweighted): 0.4726

Figure 5: Classification Report for Low Bias DNN

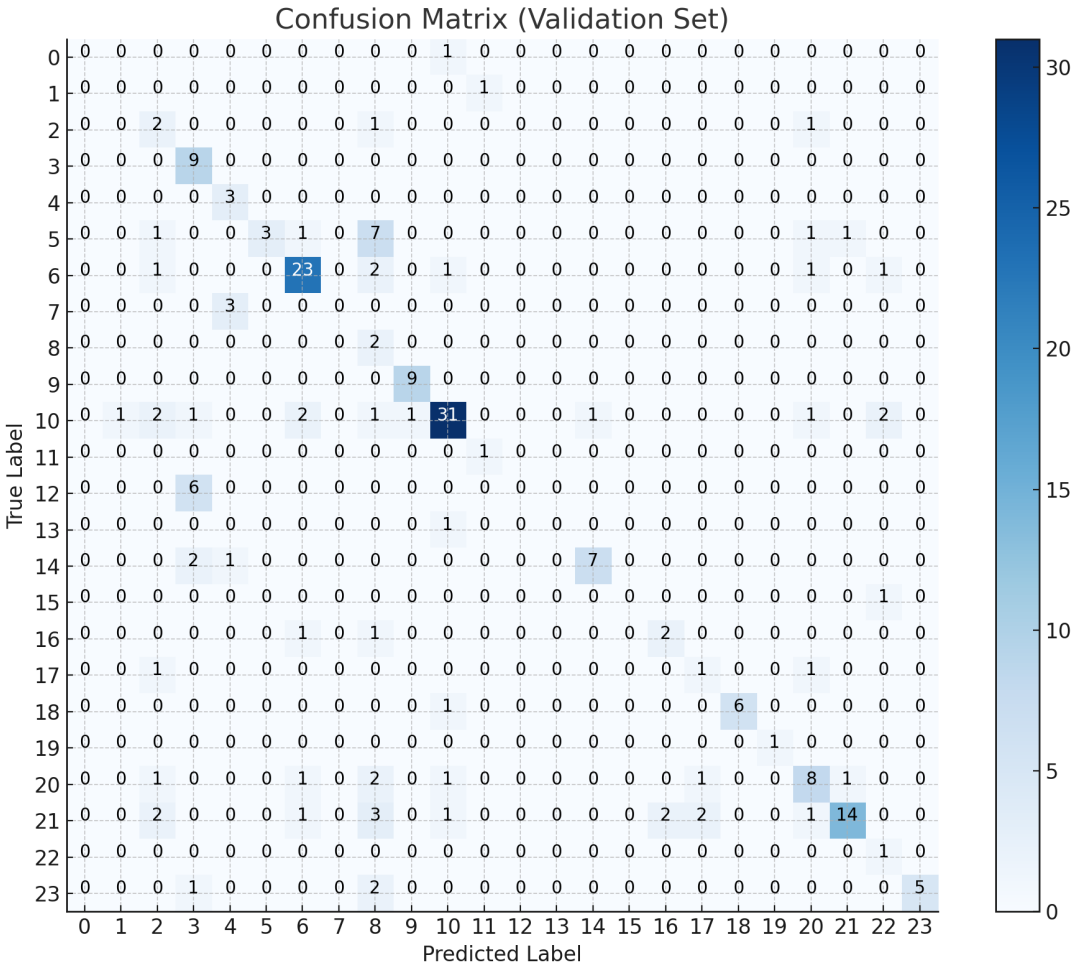

Figure 6: Confusion Matrix for Low Bias DNN Model on Test Set

Classification Report:

	precision	recall	f1-score	support
2	0.00	0.00	0.00	1
3	0.00	0.00	0.00	1
4	0.20	0.50	0.29	4
5	0.47	1.00	0.64	9
6	0.43	1.00	0.60	3
7	1.00	0.21	0.35	14
8	0.79	0.79	0.79	29
9	0.00	0.00	0.00	3
10	0.10	1.00	0.17	2
11	0.90	1.00	0.95	9
12	0.84	0.72	0.78	43
13	0.50	1.00	0.67	1
14	0.00	0.00	0.00	6
15	0.00	0.00	0.00	1
17	0.88	0.70	0.78	10
18	0.00	0.00	0.00	1
19	0.50	0.50	0.50	4
20	0.25	0.33	0.29	3
21	1.00	0.86	0.92	7
23	1.00	1.00	1.00	1
24	0.57	0.53	0.55	15
25	0.88	0.54	0.67	26
26	0.20	1.00	0.33	1
27	1.00	0.62	0.77	8
accuracy			0.63	202
macro avg	0.48	0.55	0.46	202
weighted avg	0.73	0.63	0.64	202

F1 Score (Weighted): 0.6422
F1 Score (Unweighted): 0.4602

Figure 7: Classification Report for Low Bias DNN Model on Test Set