CS412 Machine Learning
Final Project Report (Fall 2024)

Group Members: Kerim Taşkıran (29520), Eren Yamak (31283), İlke Öncü (28930)

Introduction

This report examines two key tasks in Instagram Influencer: influencer category classification and like count prediction. We address class imbalance with SMOTE, extract textual features using TF-IDF, and compare MLP and logistic regression models. Finally, we explore a simple baseline to estimate like counts despite their heavy-tailed distribution.

Part 1: Influencer Category Classification

1.1. Data Preparation and TF-IDF Feature Extraction

We began by collecting Instagram user data—each user's profile and their posts. For classification, each user is labeled with one of ten categories in the data annotation part of the project (e.g., food, fashion, entertainment, etc.):

category	user_id
art	243
entertainment	402
fashion	330
food	563
gaming	35
health and lifestyle	583
mom and children	164
sports	145
tech	390
travel	330

Table 1. Influencer Classification Label Counts

We extracted each user's post captions, cleaned them (lowercasing, removing URLs, punctuation, digits, etc.), and filtered out Turkish stopwords from NLTK. Then, we concatenated all captions belonging to a single user into one document, which we transformed using TF-IDF. TF-IDF provided a sparse vector representation, capturing how important each term is relative to the rest of the corpus. We limited ourselves to 5,000 top features for the computational cost.

Compared to raw word counts, TF-IDF downweights widespread terms and increases the importance of less common but more category-specific terms.

	abdullah	abone	about	acele	acil	activities	acı	ad	ada	adam	...	şubemiz	şubesi	şölen	şöleni	şöyle	şükranla	şükür	şık	şıklık	şıklığı
0	0.00000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.048561	0.0	0.0
1	0.00000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0
2	0.00000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.042032	0.0	0.0	0.000000	0.0	0.0
3	0.04721	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.046679	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0
4	0.00000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0
5	0.00000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.014806	0.0	0.0

Table 2. TF-IDF Text Vector Representation for the Dataset

Figure 1. Word Cloud Chart for the Most Frequent TF-IDF Words

1.2. Addressing Class Imbalance with SMOTE

One of the foremost issues was class imbalance. Certain categories—like *"health and lifestyle"* or food—had hundreds of labeled users, while others—such as *"gaming"*—had fewer than 50. Without mitigation, a classifier might be heavily biased toward the dominant classes and neglect the minority classes. To counteract this, we employed SMOTE (Synthetic Minority Oversampling Technique). SMOTE synthesizes additional samples for minority categories by interpolating between existing points in feature space, helping the classifier learn a more balanced decision boundary.

SMOTE forces the model to "see" enough examples from minority categories, improving recall and precision in classes that were previously overshadowed. However, it can lead to slight overfitting on synthetic data if not carefully validated, hence it was controlled carefully during the project.

1.3. Comparing MLP vs. Logistic Regression Models

This project progressed through three rounds, with different models used at each stage. Initially, a Multi-Layer Perceptron (MLP) classifier was employed in the first two rounds, followed by Logistic Regression with hyperparameter tuning in the final round.

Multi-Layer Perceptron (MLP): Rounds 1 and 2

The MLP was selected for its ability to capture non-linear relationships and learn complex patterns.

***Advantages*:**

- Non-linear modeling: MLP can model intricate relationships.
- Flexibility: With proper tuning, MLP can achieve high accuracy.

***Challenges*:**

- Overfitting: MLP is prone to overfitting, especially with imbalanced data, requiring careful tuning.
- Computational cost: Training MLP, particularly with SMOTE, is resource-intensive.
- Performance variability: Cross-validation showed moderate variability, particularly with minority classes.

Despite these challenges, MLP performed reasonably well in the early rounds, handling high-dimensional TF-IDF features.

Logistic Regression: Round 3

In the final round, we switched to Logistic Regression, which offered better interpretability, reduced computational needs, and robust performance with imbalanced data.

***Advantages*:**

- Simplicity and interpretability: Logistic Regression is easier to interpret.
- Efficiency: Fewer hyperparameters to tune, making it faster to train.
- Stability on imbalanced data: Consistent performance during cross-validation.

***Challenges*:**

- Limited non-linear capability: Logistic Regression is linear and may not capture complex relationships unless feature engineering is applied.

Using GridSearch, we optimized the regularization parameter (C) and solver type. The best configuration—C=10 and solver='liblinear'—was chosen based on cross-validation results. The MLP had moderately strong training accuracy, but logistic regression was more consistent across folds and simpler to tune.

While MLP has the potential for higher accuracy with larger and more balanced datasets, its complexity and resource demands made it less suitable for this task. Logistic Regression provided a simpler and more practical solution, delivering stable and interpretable results aligned with the project's conditions.

1.4. Cross-Validation and Potential Overfitting

To avoid relying solely on a single train/validation split, we used cross-validation (5-fold) on the training data. By using different data chunks for validation, we can detect overfitting in the model's performance. Indeed, our cross-validation results for logistic regression hovered around ~64–66% mean accuracy.

Signs of Overfitting:

Logistic regression's training accuracy and the validation accuracy had significant discrepancies.
This gap hints at some overfitting, where the model memorizes training patterns but cannot replicate that success on unseen data.
Still, the cross-validation approach kept that overfitting in check, ensuring we recognized the realistic performance limit.

1.5. Classification Results and Visualizations

After training and tuning our Logistic Regression model (and comparing it to the previously used MLP), we generated a variety of detailed reports and plots to fully understand how well the classifier performed and where it struggled.

Accuracy: 0.9857397504456328

Classification Report:

	precision	recall	f1-score	support
art	1.00	0.98	0.99	162
entertainment	1.00	0.97	0.98	265
fashion	0.98	1.00	0.99	245
food	0.98	1.00	0.99	413
gaming	1.00	1.00	1.00	13
health and lifestyle	0.99	0.98	0.98	413
mom and children	0.99	0.99	0.99	117
sports	1.00	1.00	1.00	98
tech	0.98	0.99	0.99	285
travel	0.98	0.98	0.98	233
accuracy			0.99	2244
macro avg	0.99	0.99	0.99	2244
weighted avg	0.99	0.99	0.99	2244

Table 3. Training Classification Report

Accuracy: 0.6601423487544484

Classification Report:

	precision	recall	f1-score	support
art	0.48	0.29	0.36	41
entertainment	0.49	0.55	0.51	66
fashion	0.65	0.59	0.62	61
food	0.77	0.88	0.83	104
gaming	0.00	0.00	0.00	3
health and lifestyle	0.59	0.72	0.65	103
mom and children	0.65	0.37	0.47	30
sports	0.88	0.58	0.70	24
tech	0.72	0.79	0.75	71
travel	0.77	0.68	0.72	59
accuracy			0.66	562
macro avg	0.60	0.54	0.56	562
weighted avg	0.66	0.66	0.65	562

Table 4. Validation Classification Report

Training vs. Validation Performance

As indicated by the **classification report on the training set** (see **Table 3**), the model achieves an exceptionally high accuracy (close to 98–99%) and near-perfect precision, recall, and F1-scores across most categories. This signals that the classifier is highly capable of memorizing or fitting the training data, especially after SMOTE has artificially balanced the categories. However, on the **validation set**, the accuracy drops to around 64–66%. This discrepancy suggests **overfitting**, since the model performance on unseen data does not match the near-perfect training metrics.

Nonetheless, 64–66% accuracy across ten categories shows a good model, given the presence of overlapping content across categories (e.g., users who occasionally post about both *food* and *health and lifestyle*, or *art* and *fashion*). It also reflects the uneven nature of real-world Instagram data, where some classes inherently have more varied text or fewer samples (e.g., *gaming*).

Confusion Matrix and Category Overlaps

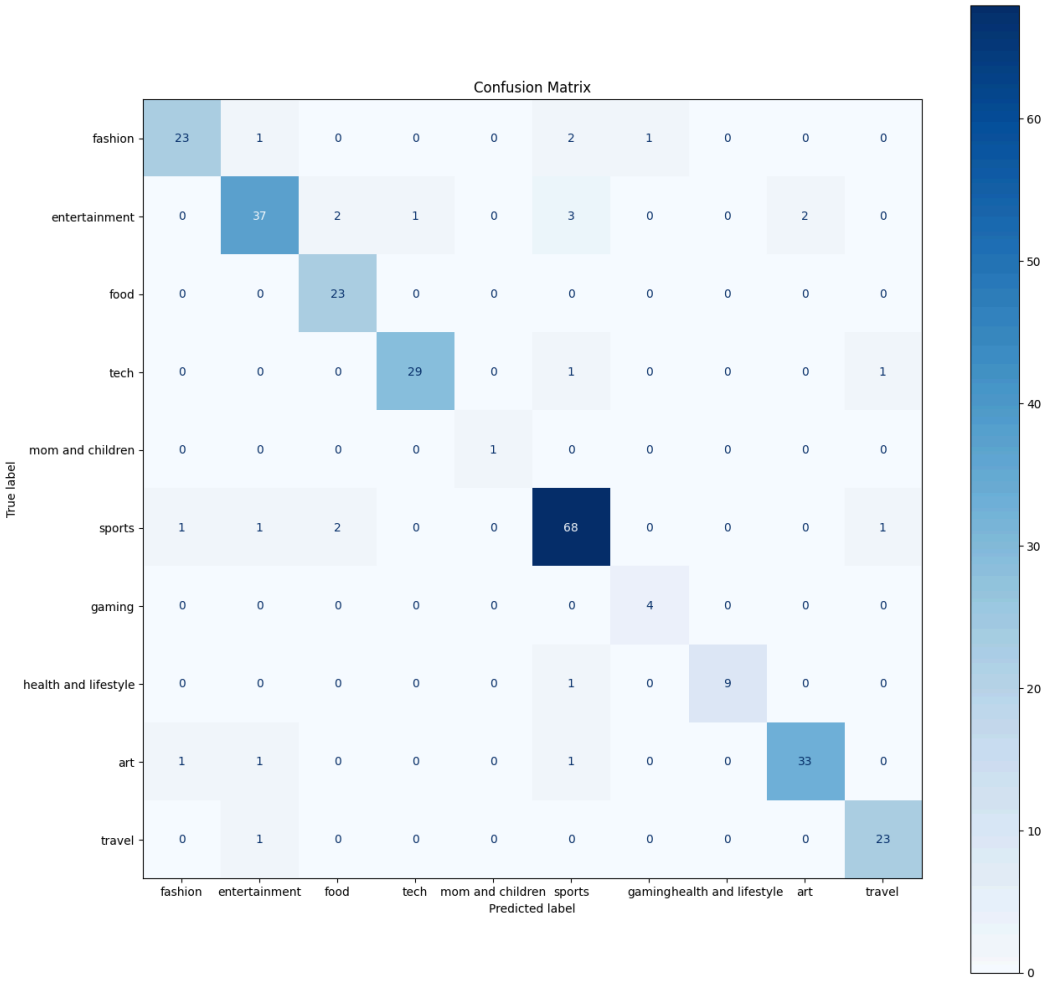

Figure 4. Confusion Matrix on Labels (Test Data)

A **confusion matrix** provides a visual breakdown of how frequently each true label aligns with a predicted label. Here are a few notable observations:

- **Sports**: It is often classified correctly (dark diagonal cell), implying that *sport*-related terminology may be quite distinct (e.g., sport branches, e-sport, etc.).
- **Gaming**: Having a very small number of training samples leads to sparse representation in TF-IDF space, causing the model to misclassify or skip predicting this category in many cases.

By studying off-diagonal cells, we see precisely which categories get mixed up, informing potential data augmentation or text-based feature refinement.

Final Test Set Observations

```
***** Test Data Accuracy *****
Accuracy: 91.24%

Classification Report:
                     precision    recall  f1-score   support

                art       0.92      0.85      0.88        27
      entertainment       0.90      0.82      0.86        45
            fashion       0.85      1.00      0.92        23
               food       0.97      0.94      0.95        31
             gaming       1.00      1.00      1.00         1
health and lifestyle       0.89      0.93      0.91        73
    mom and children       0.80      1.00      0.89         4
             sports       1.00      0.90      0.95        10
               tech       0.94      0.92      0.93        36
             travel       0.92      0.96      0.94        24

           accuracy                           0.91       274
          macro avg       0.92      0.93      0.92       274
       weighted avg       0.91      0.91      0.91       274
```

Table 5. Test Classification Report

We further tested the model on a smaller, labeled subset of the official test data (274 users). As seen in **Table 5**, the accuracy there rises to about 91.24%, and most categories exhibit high precision and recall. While this performance exceeds the validation set accuracy, it may be explained by differences in how the test subset was composed or by the fact that certain “difficult” users in the validation set are not reflected in the final test subset.

Two additional visualizations shed light on class-by-class performance:

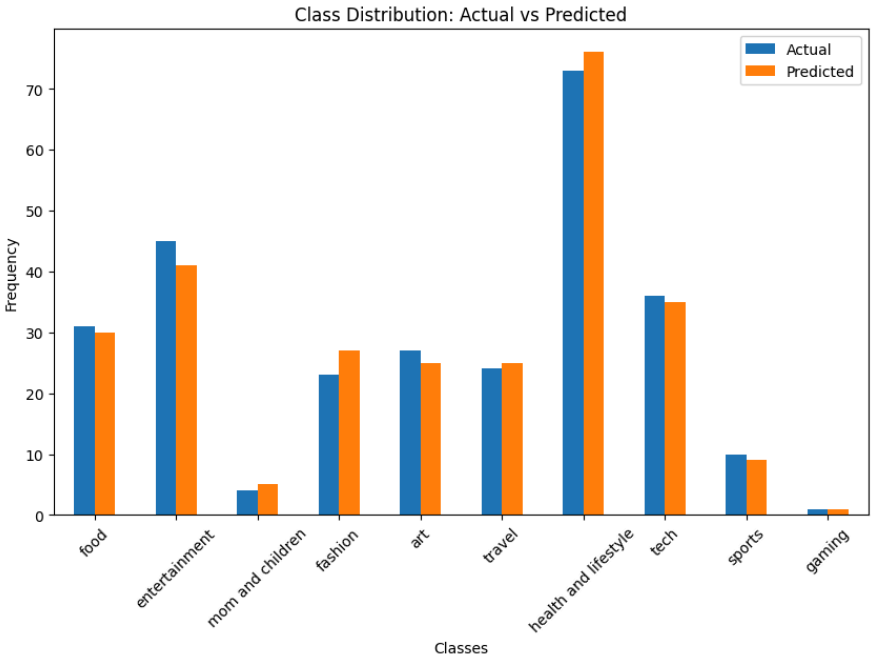

Figure 2. Class Distribution (Actual vs. Predicted)

This figure displays how many users truly belong to each category vs. how many the model predicts. Most bars align closely, though small discrepancies confirm occasional misclassifications.

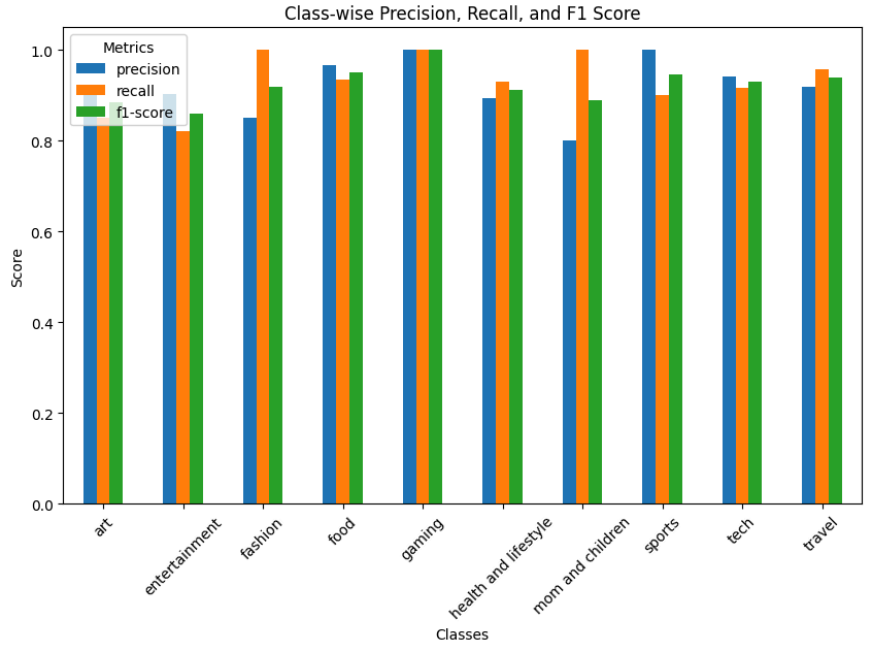

Figure 3. Class-wise Precision, Recall, and F1

The bar chart shows that some categories (e.g., *food*, *tech*, *fashion*) maintain high metrics across the board, whereas others lag due to limited samples or conceptual overlap (e.g., *mom and children*).

Overall, these test results suggest that when encountering more "typical" influencer accounts, ones that more distinctly match the text patterns learned by the model, Logistic Regression can accurately infer their categories. However, outliers or "mixed" content accounts can still pose difficulties.

Part 2: Like Count Prediction (Regression)

2.1. Heavy-Tailed Distribution and Diverse Account Popularity

For the regression task—predicting how many likes a post will receive—we face a heavy-tailed distribution: most posts land in a moderate range, but certain "viral" posts skyrocket to tens or even hundreds of thousands of likes. Coupled with the diverse popularity across accounts (some with a few hundred followers, others with hundreds of thousands), purely linear or average-based assumptions can be fragile. Large values can drastically distort error metrics.

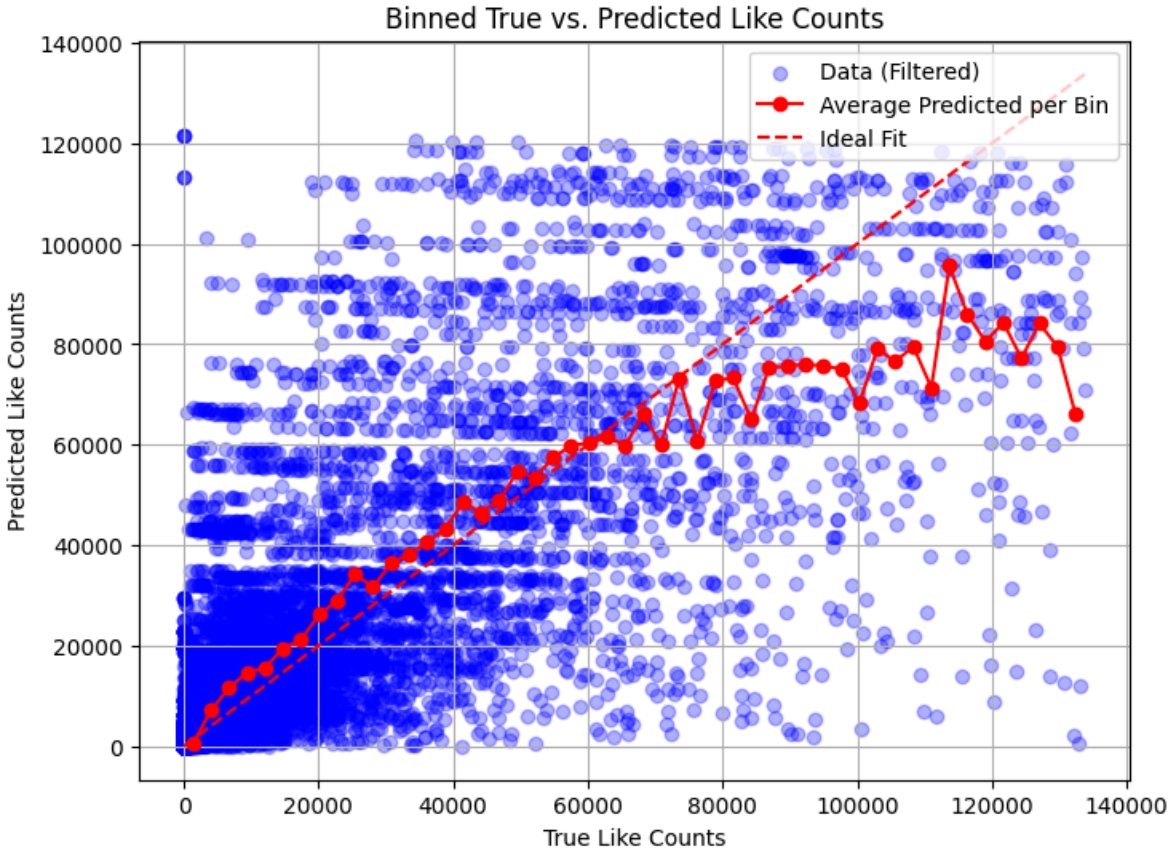

Figure 4. Binned True vs. Predicted Like Counts (with Average Line)

The scatter plot depicts each post's actual likes (x-axis) and predicted likes (y-axis). The blue dots represent individual observations, while the red line marks the average predicted values per bin of true-like counts. The dashed diagonal indicates perfect agreement between actual and predicted. At lower ranges, these predictions align closely with the ideal. At higher counts, the model often underestimates or overshoots, highlighting the challenges of heavy-tailed engagement. Overall, this visual demonstrates the usefulness of binning to compare typical vs. outlier behavior, exposing areas where the baseline predictor performs well and where it fails at capturing unexpectedly large spikes.

2.2. Baseline: Historical Average Likes

As a starting solution, we used a simple baseline: for each user, take the average of their historical like counts and apply that as the prediction for new posts. Conceptually, if a user typically receives ~500 likes, we predict ~500 for their subsequent posts. This method ignores factors like post content, hashtags, or time-of-day, but it quickly captures a user's general level of engagement.

Positive:

- Low complexity, minimal overhead, easy to interpret.
- Often surprisingly effective when user patterns are consistent.

Negative:

- Fails if a user's popularity changes or if they have widely varying engagement from post to post.
- Does not capture cyclical or event-driven spikes (holidays, collaborations, viral trends).

2.3. Evaluation via Residuals and Log MSE

We used a log-based mean squared error (Log MSE) to reduce the disproportionate influence of extremely high like counts. Specifically, we compute log(like_count + 1) for both actual and predicted values, then measure the MSE.

Training Log MSE: ~1.2156.

This indicates that on average, the (logged) predicted likes are somewhat close to the (logged) true likes, though outliers can inflate the error.

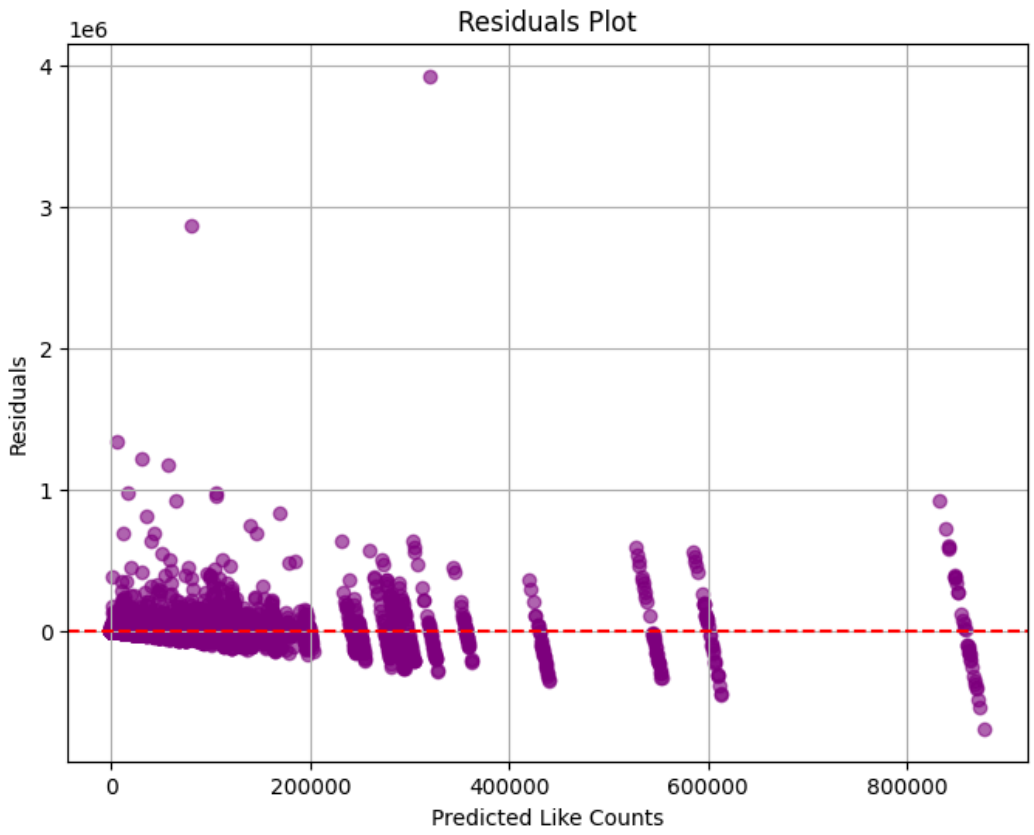

Figure 5. Residuals Plot for Like Counts

This residuals plot shows the difference between actual and predicted likes on the y-axis, with predicted likes on the x-axis. The red dashed line at zero indicates perfect predictions (no error). Points above that line are under-predictions, while points below it are over-predictions. Most data clusters below 200k predicted likes, suggesting relatively modest errors. However, some points exhibit large positive residuals, indicating substantially underestimated likes for certain popular posts. Conversely, negative residuals for higher predicted values hint that the model sometimes overestimates engagement. Overall, the spread of points reveals how well (or poorly) the model captures varying degrees of popularity.

Conclusion

Overall, these findings underscore the complexity of social media data and the need for careful consideration of both classification and regression challenges. TF-IDF text representations, combined with SMOTE, significantly improved the classifier's ability to handle underrepresented classes. While the MLP captured non-linearities, logistic regression ultimately offered more consistent performance with fewer hyperparameters. The like count prediction component highlighted how diverse account popularity and heavy-tailed engagement pose unique difficulties. Our baseline approach, which predicts a user's average likes, delivered a decent starting point, but deeper modeling may be required to capture viral spikes or low-performing anomalies. Future refinements could yield even more robust insights.

Appendix

ROC Curve for Classification (Part 1)