Feature Engineering for SVM, Logistic Regression & Decision Tree

Soren Larsen
UCSC Silicon Valley Extension Campus
Santa Clara, California
snlarsen@ucsc.com

Abstract

This report details the completion of NLP 220 Assignment 2, which involves feature engineering for SVM, Logistic Regression, and Decision Tree classifiers on an e-commerce dataset. The assignment explores classification, feature engineering, model evaluation metrics, hyperparameter exploration, and OneVsRest comparisons for a 4-class problem.

1 Introduction

This assignment aims to implement and evaluate feature engineering techniques for SVM, Logistic Regression, and Decision Tree classifiers on an e-commerce dataset. The goal is to optimize models for a multi-class classification task involving item categories.

2 Dataset and Preprocessing

The dataset contains categories and descriptions of items from an e-commerce website. A train/test split is created using 70% training, 10% validation, and 20% test data, ensuring reproducibility with a fixed random seed.

2.1 Class Distribution

To better understand the dataset, the distribution of classes/labels is plotted and analyzed. (See Appendix O).

3 Feature Engineering and Modeling

For this multi-class classification problem, three distinct feature engineering techniques were applied to SVM, Logistic Regression, and Decision Tree classifiers, resulting in a total of nine models.

3.1 Feature Descriptions

The following feature engineering techniques were applied:

- **GloVe Embeddings:** Average word embeddings were generated using the pre-trained `glove-wiki-gigaword-300` model from Gensim. Each item description was converted to a fixed-dimensional embedding by averaging over word vectors, effectively capturing semantic similarities in descriptions.

- **Sublinear TF-IDF:** Sublinear Term Frequency-Inverse Document Frequency (TF-IDF) was used to emphasize less frequent terms. This feature extraction technique created weighted term vectors for item descriptions, with sublinear scaling applied to term frequencies, highlighting unique terms.

- **Combined NLTK and Bag-of-Words Features:** In this setup, linguistic features were combined with Bag-of-Words representations:
  - **Named Entity Count:** Extracted using NLTK's `ne_chunk` function.
  - **POS Ratios:** Ratios of nouns, verbs, and adjectives were calculated.
  - **Sentiment Score:** Generated with NLTK's `SentimentIntensityAnalyzer`.
  - **Basic Counts:** Word and character counts.

  The combined NLTK and Bag-of-Words features were then standardized using `StandardScaler` to improve model performance and ensure consistent scale across features.

4 Training and Inference Time

The training time for each classifier and feature set combination was recorded and compared to evaluate the computational efficiency of the models. Additionally, the preprocessing time for each feature engineering technique was documented.

4.1 Preprocessing and Training Times by Feature Engineering Technique

- **GloVe Embeddings:**
  - **Embedding Loading and Feature Extraction Time:** 27.68 seconds
  - Logistic Regression Training Time: 20.70 seconds
  - SVM Training Time: 1 minute 14.06 seconds
  - Decision Tree Training Time: 29.51 seconds
- **Sublinear TF-IDF:**
  - **TF-IDF Extraction Time:** 1.49 seconds
  - Logistic Regression Training Time: 11.68 seconds
  - SVM Training Time: 2.84 seconds
  - Decision Tree Training Time: 27.93 seconds
- **Combined NLTK and Bag-of-Words Features:**
  - **Bag-of-Words Extraction Time:** 1.48 seconds
  - **Parallel NLTK Feature Extraction Time:** 15 minutes 31.92 seconds
  - **Feature Scaling Time:** 0.20 seconds
  - Logistic Regression Training Time: 30.07 seconds
  - SVM Training Time: 22 minutes 51.08 seconds
  - Decision Tree Training Time: 21.76 seconds

5 Hyperparameter Descriptions

For each classifier, I explored different values for key hyperparameters:

- **C (Regularization Parameter):** Used in Logistic Regression and SVM, this parameter controls the regularization strength. Smaller values indicate stronger regularization, reducing overfitting but potentially underfitting the model.

- **max_depth:** In the Decision Tree classifier, this parameter sets the maximum depth of the tree. Limiting the depth helps prevent overfitting by constraining the complexity of the model.

- **min_samples_split:** This Decision Tree parameter defines the minimum number of samples required to split an internal node. Higher values prevent the tree from learning overly specific patterns by limiting node creation.

6 Model Evaluation with GloVe Embeddings

This section presents the evaluation results for three classifiers—Logistic Regression, Support Vector Machine (SVM), and Decision Tree—using GloVe Embeddings as features. Each model was optimized through hyperparameter tuning, and results are provided below.

6.1 Logistic Regression with GloVe Embeddings

Best Hyperparameters: $C = 10$
Training Time: 20.7 seconds

Validation Performance

- **Accuracy:** 0.8096
- **Macro F1:** 0.8055

Test Performance

- **Accuracy:** 0.7974
- **Macro F1:** 0.7921

Table 1: Validation Performance for Logistic Regression with GloVe Embeddings

Parameter (C)	Fold 1 F1	Fold 2 F1	Fold 3 F1
0.01	0.7827	0.7811	0.7780
0.1	0.8021	0.7951	0.7917
1	0.8044	0.7968	0.7985
10	0.8078	0.7964	0.8012
100	0.8054	0.7972	0.8018

The confusion matrix for this model on the test set is provided in Appendix A.

6.2 SVM with GloVe Embeddings

Best Hyperparameters: $C = 100$
Training Time: 1 minute 14.06 seconds

Validation Performance

- **Accuracy:** 0.8148
- **Macro F1:** 0.8099

Test Performance

- **Accuracy:** 0.8066
- **Macro F1:** 0.8012

Table 2: Validation Performance for SVM with GloVe Embeddings

Parameter (C)	Fold 1 F1	Fold 2 F1	Fold 3 F1
0.01	0.8014	0.8022	0.7985
0.1	0.8075	0.8052	0.8062
1	0.8099	0.8070	0.8078
10	0.8093	0.8089	0.8088
100	0.8097	0.8099	0.8086

Refer to Appendix B for the SVM confusion matrix on the test set.

6.3 Decision Tree with GloVe Embeddings

Best Hyperparameters: `max_depth = 20`, `min_samples_split = 2`
Training Time: 29.51 seconds

Validation Performance

- **Accuracy:** 0.8384
- **Macro F1:** 0.8313

Test Performance

- **Accuracy:** 0.8251
- **Macro F1:** 0.8169

Table 3: Validation Performance for Decision Tree with GloVe Embeddings (Parameters: max_depth, min_samples_split)

Parameter	Fold 1 F1	Fold 2 F1	Fold 3 F1
None, 2	0.7906	0.7901	0.7882
None, 5	0.7834	0.7855	0.7803
None, 10	0.7705	0.7766	0.7702
10, 2	0.7509	0.7503	0.7469
10, 5	0.7497	0.7501	0.7460

The confusion matrix for Decision Tree on the test set is provided in Appendix C.

7 Model Evaluation with Sublinear TF-IDF

This section presents the evaluation results for three classifiers—Logistic Regression, Support Vector Machine (SVM), and Decision Tree—using Sublinear TF-IDF features. Each model was optimized through hyperparameter tuning, and results are presented below.

7.1 Logistic Regression with Sublinear TF-IDF

Best Hyperparameters: $C = 1$
Training Time: 11.68 seconds

Validation Performance

- **Accuracy:** 0.9449
- **Macro F1:** 0.9458

Test Performance

- **Accuracy:** 0.9355
- **Macro F1:** 0.9365

Table 4: Validation Performance for Logistic Regression with Sublinear TF-IDF

Parameter (C)	Fold 1 F1	Fold 2 F1	Fold 3 F1
0.01	0.9116	0.9146	0.9083
0.1	0.9327	0.9319	0.9273
1	0.9407	0.9407	0.9377
10	0.9393	0.9380	0.9366
100	0.9302	0.9338	0.9285

The confusion matrix for this model on the test set is provided in Appendix D.

7.2 SVM with Sublinear TF-IDF

Best Hyperparameters: $C = 1$
Training Time: 2.84 seconds

Validation Performance

- **Accuracy:** 0.9427
- **Macro F1:** 0.9436

Test Performance

- **Accuracy:** 0.9352
- **Macro F1:** 0.9358

Refer to Appendix E for the SVM confusion matrix on the test set.

Table 5: Validation Performance for SVM with Sublinear TF-IDF

Parameter (C)	Fold 1 F1	Fold 2 F1	Fold 3 F1
0.01	0.9295	0.9310	0.9243
0.1	0.9408	0.9401	0.9379
1	0.9408	0.9421	0.9392
10	0.9357	0.9383	0.9342
100	0.9320	0.9336	0.9312

7.3 Decision Tree with Sublinear TF-IDF

Best Hyperparameters: `max_depth = None`, `min_samples_split = 2`
Training Time: 27.93 seconds

Validation Performance

- **Accuracy:** 0.9391
- **Macro F1:** 0.9401

Test Performance

- **Accuracy:** 0.9369
- **Macro F1:** 0.9379

Table 6: Validation Performance for Decision Tree with Sublinear TF-IDF

Parameter	Fold 1 F1	Fold 2 F1	Fold 3 F1
None, 2	0.9147	0.9129	0.9142
None, 5	0.9117	0.9080	0.9096
None, 10	0.9044	0.9008	0.9022
10, 2	0.7343	0.7433	0.7410
10, 5	0.7342	0.7430	0.7409

The confusion matrix for Decision Tree on the test set is provided in Appendix F.

8 Combined NLTK and Bag-of-Words Features

In this setup, Bag-of-Words representations were combined with NLTK-based linguistic features, followed by feature scaling to standardize the data. The feature extraction process involved:

- **Bag-of-Words Extraction:** Completed in 1.48 seconds, providing a vector representation of text based on word occurrence counts.
- **Parallel NLTK Feature Extraction:** Extracted using parallel processing in 15 minutes and 31.92 seconds, including named entity counts, POS ratios, sentiment scores, and basic word/character counts.
- **Feature Scaling:** Standardized features using `StandardScaler`, completed in 0.20 seconds to improve model convergence and accuracy.

8.1 Logistic Regression Analysis

Using the combined NLTK and Bag-of-Words features, multiple values of regularization parameter C were explored.

Hyper-parameter Tuning Results:

- **Parameters: {C: 0.01}**
  Mean F1 Macro Score: 0.9270 (+/- 0.0018)
- **Parameters: {C: 0.1}**
  Mean F1 Macro Score: 0.9253 (+/- 0.0032)
- **Parameters: {C: 1}**
  Mean F1 Macro Score: 0.9189 (+/- 0.0032)
- **Parameters: {C: 10}**
  Mean F1 Macro Score: 0.9139 (+/- 0.0024)
- **Parameters: {C: 100}**
  Mean F1 Macro Score: 0.9133 (+/- 0.0025)

Best Hyperparameters: $C = 0.01$
Training Time: 30.07 seconds

Validation Performance

- **Accuracy:** 0.9320
- **Macro F1:** 0.9338

Test Performance

- **Accuracy:** 0.9217
- **Macro F1:** 0.9231

The confusion matrix for the Logistic Regression model on the test set is provided in Appendix G.

8.2 SVM Analysis

Using the combined NLTK and Bag-of-Words features, hyper-parameter tuning was conducted for SVM with multiple values of C.

Hyper-parameter Tuning Results:

- **Parameters: {C: 0.01}**
  Mean F1 Macro Score: 0.9285 (+/- 0.0011)
- **Parameters: {C: 0.1}**
  Mean F1 Macro Score: 0.9264 (+/- 0.0017)
- **Parameters: {C: 1}**
  Mean F1 Macro Score: 0.9235 (+/- 0.0008)
- **Parameters: {C: 10}**
  Mean F1 Macro Score: 0.9205 (+/- 0.0012)
- **Parameters: {C: 100}**
  Mean F1 Macro Score: 0.9202 (+/- 0.0012)

Best Hyperparameters: $C = 0.01$
Training Time: 22 minutes and 51.08 seconds

Validation Performance

- **Accuracy:** 0.9322
- **Macro F1:** 0.9335

Test Performance

- **Accuracy:** 0.9232
- **Macro F1:** 0.9241

The confusion matrix for the SVM model on the test set is provided in Appendix H.

8.3 Decision Tree Analysis

Hyper-parameter tuning for the Decision Tree classifier was conducted using the combined NLTK and Bag-of-Words features.

Hyper-parameter Tuning Results:

- **Parameters: {max_depth: None, min_samples_split: 2}**
  Mean F1 Macro Score: 0.9131 (+/- 0.0014)
- **Parameters: {max_depth: None, min_samples_split: 5}**
  Mean F1 Macro Score: 0.9081 (+/- 0.0036)
- **Parameters: {max_depth: None, min_samples_split: 10}**
  Mean F1 Macro Score: 0.9041 (+/- 0.0029)
- **Parameters: {max_depth: 10, min_samples_split: 2}**
  Mean F1 Macro Score: 0.7886 (+/- 0.0031)
- **Parameters: {max_depth: 20, min_samples_split: 2}**
  Mean F1 Macro Score: 0.8552 (+/- 0.0038)

Best Hyperparameters: `max_depth = None`, `min_samples_split = 2`
Training Time: 21.76 seconds

Validation Performance

- **Accuracy:** 0.9381
- **Macro F1:** 0.9396

Test Performance

- **Accuracy:** 0.9376
- **Macro F1:** 0.9380

The confusion matrix for the Decision Tree model on the test set is provided in Appendix I.

9 Model Comparison and Discussion

The performance of the nine model-feature combinations demonstrated notable differences due to variations in feature representation and model architecture. The test set accuracy and macro-average F1 scores for each combination are summarized as follows:

Table 7: Test Performance Summary (Accuracy and Macro F1)

Model-Feature Combination	Accuracy	Macro F1
Logistic Regression + GloVe	0.7974	0.7921
SVM + GloVe	0.8066	0.8012
Decision Tree + GloVe	0.8251	0.8169
Logistic Regression + TF-IDF	0.9355	0.9365
SVM + TF-IDF	0.9352	0.9358
Decision Tree + TF-IDF	0.9369	0.9379
Logistic Regression + NLTK-BoW	0.9217	0.9231
SVM + NLTK-BoW	0.9232	0.9241
Decision Tree + NLTK-BoW	0.9376	0.9380

The models using Sublinear TF-IDF consistently achieved the highest accuracy and macro F1 scores across all classifiers. This can be attributed to TF-IDF's ability to emphasize informative and discriminative terms within the dataset, providing more meaningful representations for classification tasks.

GloVe Embeddings exhibited lower performance across all models, with Logistic Regression achieving the lowest accuracy (0.7974) and macro F1 score (0.7921). This can be explained by GloVe's semantic focus, which may have captured less discriminative detail relevant for distinguishing among the four e-commerce categories.

The **Combined NLTK and Bag-of-Words** features led to strong performance, particularly for Decision Tree (accuracy: 0.9376, macro F1: 0.9380) and SVM (accuracy: 0.9232, macro F1: 0.9241). The inclusion of linguistic features such as named entity counts and POS ratios, combined with Bag-of-Words, offered a richer representation, which was better exploited by non-linear models like Decision Trees.

10 Timing Comparison and Analysis

The feature extraction and model training times varied significantly due to differences in feature complexity and model structure. Table 17 highlights the timing comparisons (refer to Appendix N for more details).

Feature Extraction Time: Sublinear TF-IDF had the shortest extraction time (1.49 sec) due to its straightforward computation of term weights. In contrast, the NLTK and Bag-of-Words combination took over 15 minutes due to complex linguistic feature extraction, including named entity recognition and POS tagging.

Model Training Time: Logistic Regression trained quickly across all feature sets due to its linear nature. SVM exhibited higher training times for Combined NLTK and Bag-of-Words features due to kernel computations in a high-dimensional feature space. Decision Trees maintained efficient training times due to their recursive structure, even when handling complex features.

11 Best Hyperparameter Configurations

The optimal hyperparameter configurations for each model-feature combination were as follows:

- **Logistic Regression with GloVe Embeddings:** $C = 10$
- **SVM with GloVe Embeddings:** $C = 100$
- **Decision Tree with GloVe Embeddings:** max_depth = 20, min_samples_split = 2
- **Logistic Regression with Sublinear TF-IDF:** $C = 1$
- **SVM with Sublinear TF-IDF:** $C = 1$
- **Decision Tree with Sublinear TF-IDF:** max_depth = None, min_samples_split = 2
- **Logistic Regression with Combined NLTK and Bag-of-Words Features:** $C = 0.01$
- **SVM with Combined NLTK and Bag-of-Words Features:** $C = 0.01$
- **Decision Tree with Combined NLTK and Bag-of-Words Features:** max_depth = None, min_samples_split = 2

12 OneVsRest Exploration

This section describes the evaluation of a OneVsRest approach for a 4-class classification problem using a Decision Tree model optimized with Sublinear TF-IDF features. The OneVsRest strategy involves treating each class as the positive class while considering all other classes as the negative class, effectively producing four different binary classification scenarios. For each class, individual OneVsRest accuracy and macro-average F1 scores were computed to assess the classifier's performance.

12.1 ROC and Precision-Recall Curves

For each class, both ROC and Precision-Recall curves were generated to visualize the model's discriminative performance. The curves for each class are presented in the appendix.

12.1.1 Class 1: Clothing & Accessories

- **ROC Curve:** The ROC curve for the Clothing & Accessories class demonstrates a strong ability of the classifier to distinguish positive examples from negative ones, as indicated by a curve that approaches the top-left corner. This suggests a high True Positive Rate with a relatively low False Positive Rate. Refer to Appendix J.
- **Precision-Recall Curve:** The Precision-Recall curve shows high precision at high recall values, indicating that most of the positive predictions for this class are accurate, with a slight drop-off in precision as recall increases. Refer to Appendix J.

12.1.2 Class 2: Electronics

- **ROC Curve:** The ROC curve for the Electronics class is close to the ideal diagonal, demonstrating a reasonably high True Positive Rate, but with slightly more false positives than the

Clothing & Accessories class. Refer to Appendix K.

- **Precision-Recall Curve:** The Precision-Recall curve maintains high precision for most recall values, but a drop-off is observed as recall approaches maximum. This suggests a need to carefully balance precision and recall for this class. Refer to Appendix K.

12.1.3 Class 3: Household

- **ROC Curve:** The ROC curve for the Household class demonstrates relatively good discriminative performance, with a curve that rises quickly towards the top-left corner. This indicates effective classification, albeit with a small degree of false positives. Refer to Appendix L.

- **Precision-Recall Curve:** The Precision-Recall curve indicates consistently high precision at most levels of recall. A steep drop-off at high recall values suggests some difficulty in maintaining precision as more instances are identified as positive. Refer to Appendix L.

12.1.4 Class 4: Books

- **ROC Curve:** The ROC curve for the Books class is also close to the top-left corner, indicating strong performance in distinguishing positive instances from negative ones. Refer to Appendix M.

- **Precision-Recall Curve:** The Precision-Recall curve shows high precision for most recall values but includes a noticeable drop as recall increases. This suggests a robust performance for identifying positive instances but with a trade-off in precision when more instances are considered. Refer to Appendix M.

12.2 Conclusion

The OneVsRest evaluation demonstrated strong performance for each class using the Decision Tree model optimized with Sublinear TF-IDF features. ROC and Precision-Recall curves for each class highlight the strengths and weaknesses in the classifier's ability to separate positive and negative instances, providing valuable insights for future model tuning and feature engineering.

Appendix

A Logistic Regression with GloVe Embeddings

Table 8: Confusion Matrix for Logistic Regression with GloVe Embeddings

	Elect.	**House.**	**Books**	**Clothing**
Elect.	1892	288	85	99
House.	93	1490	64	87
Books	103	124	1697	200
Clothing	211	331	358	2963

B SVM with GloVe Embeddings

Table 9: Confusion Matrix for SVM with GloVe Embeddings

	Elect.	**House.**	**Books**	**Clothing**
Elect.	2130	49	63	122
House.	264	1310	48	112
Books	192	54	1614	264
Clothing	319	187	276	3081

C Decision Tree with GloVe Embeddings

Table 10: Confusion Matrix for Decision Tree with GloVe Embeddings

	Elect.	**House.**	**Books**	**Clothing**
Elect.	1951	272	54	87
House.	87	1403	52	192
Books	76	109	1733	206
Clothing	153	260	216	3234

D Logistic Regression with Sublinear TF-IDF

Table 11: Confusion Matrix for Logistic Regression with Sublinear TF-IDF

	Elect.	House.	Books	Clothing
Elect.	2228	15	38	83
House.	16	1669	13	36
Books	48	16	1957	103
Clothing	100	72	110	3581

E SVM with Sublinear TF-IDF

Table 12: Confusion Matrix for SVM with Sublinear TF-IDF

	Elect.	House.	Books	Clothing
Elect.	2220	18	36	90
House.	17	1668	14	35
Books	56	18	1935	115
Clothing	95	60	100	3608

F Decision Tree with Sublinear TF-IDF

Table 13: Confusion Matrix for Decision Tree with Sublinear TF-IDF

	Elect.	House.	Books	Clothing
Elect.	2244	9	39	72
House.	21	1642	18	53
Books	27	15	1960	122
Clothing	96	53	111	3603

G Logistic Regression with Combined NLTK and Bag-of-Words Features

Table 14: Confusion Matrix for Logistic Regression with Combined NLTK and Bag-of-Words Features

	Elect.	House.	Books	Clothing
Elect.	2243	16	40	65
House.	38	1654	9	33
Books	86	13	1910	115
Clothing	195	71	109	3488

H SVM with Combined NLTK and Bag-of-Words Features

Table 15: Confusion Matrix for SVM with Combined NLTK and Bag-of-Words Features

	Elect.	House.	Books	Clothing
Elect.	2228	18	40	78
House.	39	1650	10	35
Books	84	17	1897	126
Clothing	168	65	95	3535

I Decision Tree with Combined NLTK and Bag-of-Words Features

Table 16: Confusion Matrix for Decision Tree with Combined NLTK and Bag-of-Words Features

	Elect.	House.	Books	Clothing
Elect.	2238	16	35	75
House.	31	1624	14	65
Books	31	13	1967	113
Clothing	84	50	102	3627

J ROC and Precision-Recall Curves for Clothing & Accessories

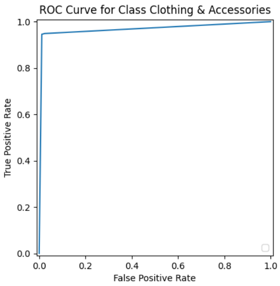

Figure 1: ROC Curve for Class Clothing & Accessories.

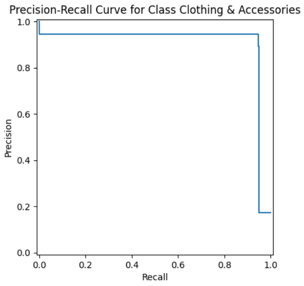

Figure 2: Precision-Recall Curve for Class Clothing & Accessories.

K ROC and Precision-Recall Curves for Electronics

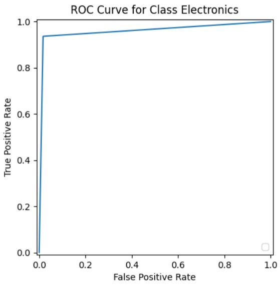

Figure 3: ROC Curve for Class Electronics.

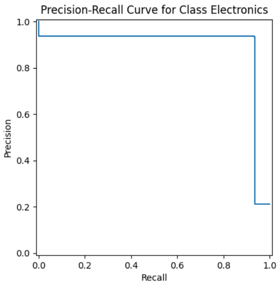

Figure 4: Precision-Recall Curve for Class Electronics.

L ROC and Precision-Recall Curves for Household

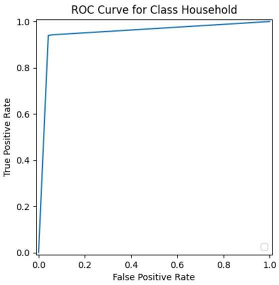

Figure 5: ROC Curve for Class Household.

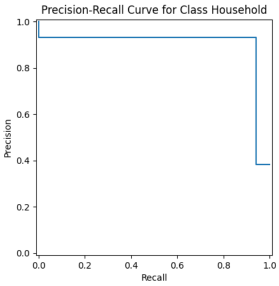

Figure 6: Precision-Recall Curve for Class Household.

M ROC and Precision-Recall Curves for Books

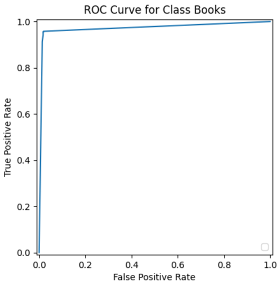

Figure 7: ROC Curve for Class Books.

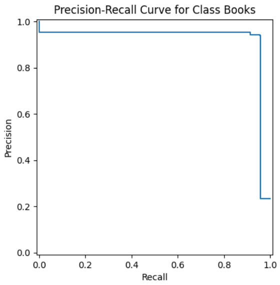

Figure 8: Precision-Recall Curve for Class Books.

N Timing Summary for Feature Extraction and Model Training

Table 17: Timing Summary (Feature Extraction and Training)

Feature Set	**Time (sec)**
GloVe Embeddings	
Extraction Time	27.68 sec
Logistic Regression (LR) Training Time	20.70 sec
Support Vector Machine (SVM) Training Time	1 min 14.06 sec
Decision Tree (DT) Training Time	29.51 sec
Sublinear TF-IDF	
Extraction Time	1.49 sec
Logistic Regression (LR) Training Time	11.68 sec
Support Vector Machine (SVM) Training Time	2.84 sec
Decision Tree (DT) Training Time	27.93 sec
NLTK + Bag-of-Words (BoW)	
Bag-of-Words Extraction Time	1.48 sec
Parallel NLTK Feature Extraction Time	15 min 31.92 sec
Feature Scaling Time	0.20 sec
Logistic Regression (LR) Training Time	30.07 sec
Support Vector Machine (SVM) Training Time	22 min 51.08 sec
Decision Tree (DT) Training Time	21.76 sec

O Class Distribution

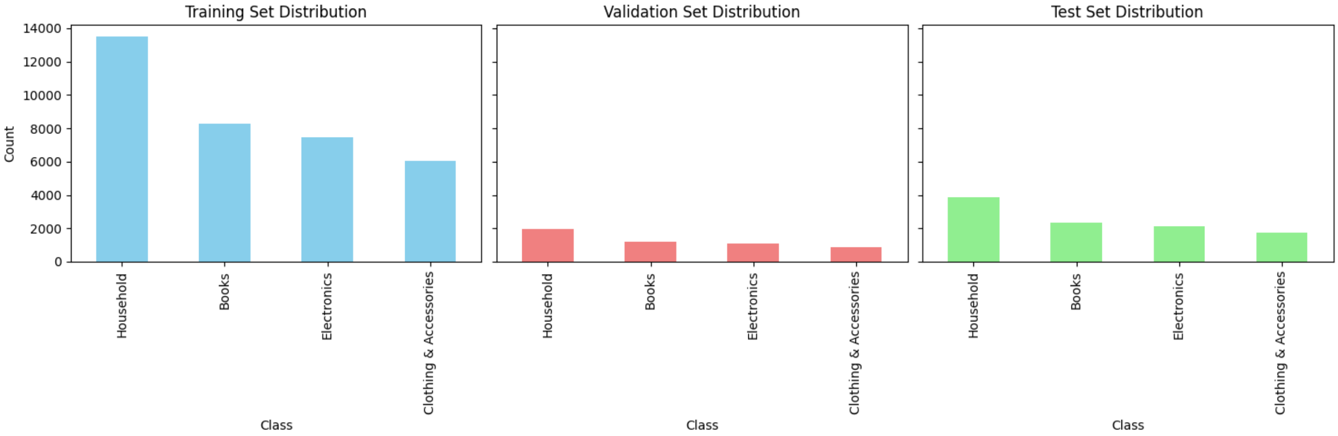

Figure 9: Class distribution across training, validation, and test sets.