Multi-class Text Classification on 20_Newsgroups Dataset

CMT122 Coursework 1 - Part 2

Samidur Rahman
Student ID: 21053329

Academic Year 2025/2026

Abstract

This report presents a machine learning pipeline for multi-class text classification on the 20_Newsgroups dataset (3,416 articles, 6 categories). Combining TF-IDF vectorization with statistical text features, chi-squared feature selection, and LinearSVC classification, the system achieved 88.74% test accuracy with macro-averaged F1-score of 0.8906. Development set experiments validated all design choices, while critical reflection addresses potential improvements and biases.

1 Methodology Description

1.1 Data Preprocessing

The dataset contains 3,416 newsgroup articles across 6 categories. Preprocessing involved converting text to lowercase, removing email addresses and headers using regex patterns (`\S+@\S+`), stripping non-alphabetic characters (`[^a-z\s]`), normalizing whitespace, and filtering documents shorter than 50 characters. This cleaning focused the model on substantive content rather than metadata or formatting artifacts.

1.2 Dataset Partitioning

Following best practices, the dataset was stratified into training (60%, 2,049 samples), development (20%, 683 samples), and test (20%, 684 samples) sets. The development set enabled hyperparameter tuning without test set contamination, while stratification maintained class proportions across splits, preventing distribution shift.

1.3 Feature Engineering

Four features captured complementary document aspects:

Feature 1: TF-IDF. Primary representation with parameters: `max_features=20000` (vocabulary size), `ngram_range=(1,2)` (unigrams and bigrams for multi-word concepts), `stop_words='english'`, `min_df=3` and `max_df=0.75` (frequency thresholds), `sublinear_tf=True` (logarithmic scaling). After filtering, this produced a 2,049 $\times$ 19,226 sparse matrix (98% sparsity).

Features 2-4: Statistical features. Word count captured document length; average word length measured vocabulary complexity; lexical diversity (type-token ratio) quantified vocabulary variety. These features were scaled to [0,1] using MinMaxScaler (fitted on training only) for chi-squared compatibility and to prevent range-based dominance. All features were concatenated via `hstack` into a 2,049 $\times$ 19,229 sparse matrix.

1.4 Feature Selection

Chi-squared testing identified discriminative features by measuring dependence between features and class labels. Development experiments tested $k \in \{1000, 5000, 10000, 15000, 19229\}$ to determine optimal feature count, with each configuration evaluated via LinearSVC on the development set.

1.5 Model Training

LinearSVC was selected for its effectiveness with high-dimensional sparse text data. Hyperparameters included `C=1.0` (regularization), `max_iter=2000`, and `random_state=42`. After identifying optimal parameters through development experiments, the final model was trained on combined training and development sets (2,732 samples) before single evaluation on the test set.

2 Justification of Design Choices

2.1 Feature Engineering Rationale

TF-IDF was selected to downweight common terms while emphasising distinctive vocabulary through IDF scaling. Bigrams captured domain-specific phrases ("computer graphics", "machine learning") absent in unigram-only models. The 20,000-feature vocabulary balanced coverage with efficiency; preliminary experiments with 5,000 features excluded important terminology, while larger vocabularies showed diminishing returns. Frequency thresholds filtered noise (rare typos via `min_df=3`) and overly common terms (`max_df=0.75`).

Statistical features complemented TF-IDF by capturing structural properties. For example, word count helped distinguish lengthy technical posts from brief conversational exchanges, average word length separated specialised terminology from casual, and lexical diversity separated original content from repetitive. These meta-features provided valuable signals to word usage patterns.

2.2 Feature Selection Experiments

Development experiments systematically evaluated feature set sizes (Table 1).

Table 1: Feature selection results on development set

k (features)	% of Total	Dev Accuracy	Reduction
1,000	5.2%	0.8477	94.8%
5,000	26.0%	0.8917	74.0%
10,000	52.0%	0.9019	48.0%
15,000	**78.0%**	**0.9034**	**22.0%**
19,229 (all)	100.0%	0.8975	0.0%

Performance peaked at $k = 15,000$ (90.34% accuracy), with all features yielding slightly lower performance (89.75%). This indicated approximately 4,000 features introduced noise rather than signal. The largest gain occurred between 5,000 and 10,000 features (+1.02pp), demonstrating that chi-squared successfully prioritised discriminative features. Aggressive reduction to 1,000 features caused 5.57pp degradation, illustrating the risk of excessive pruning.

2.3 Model Selection

Four algorithms were compared on development data using optimal features ($k = 15,000$), as shown in Table 2.

Table 2: Model comparison on development set

Model	Dev Accuracy
LinearSVC (C=1.0)	**0.9034**
LinearSVC (C=0.5)	0.9019
Logistic Regression	0.8799
Multinomial Naive Bayes	0.8858

LinearSVC with `C=1.0` achieved highest accuracy (90.34%). The linear kernel exploited linear separability common in high-dimensional TF-IDF spaces. Reduced regularization (`C=0.5`) marginally decreased performance (90.19%), confirming appropriate default parameterization. Alternative algorithms (Logistic Regression: 87.99%, Naive Bayes: 88.58%) underperformed, validating SVM's maximum-margin principle for this task.

3 Performance Results

3.1 Test Set Performance

Final model performance on held-out test data (Table 3).

Table 3: Final test set performance

Metric	Value
Accuracy	0.8874 (88.74%)
Macro-averaged Precision	0.8909
Macro-averaged Recall	0.8906
Macro-averaged F1-score	0.8906

Test accuracy of 88.74% exceeded the 65% requirement by 23.74 percentage points. Macro-averaged metrics (≈ 0.89) demonstrated balanced performance across classes, with precision-recall alignment indicating consistent quality in both identification and prediction accuracy.

3.2 Per-Class Analysis

Per-class metrics (Table 4) revealed performance variations.

Table 4: Per-class performance on test set

Class	Precision	Recall	F1-score	Support
class-1	0.99	0.99	0.99	96
class-2	0.83	0.86	0.85	117
class-3	0.86	0.87	0.86	119
class-4	0.84	0.81	0.82	118
class-5	0.93	0.90	0.92	115
class-6	0.89	0.92	0.90	119

Class-1 achieved near-perfect performance (99% F1), suggesting highly distinctive vocabulary. Classes 2 and 4 showed lowest scores (82-85% F1), indicating thematic overlap. All classes maintained F1 > 0.82, demonstrating balanced generalization without severe per-class bias.

The confusion matrix (Table 5) revealed specific misclassification patterns.

Table 5: Confusion matrix (rows: true, columns: predicted)

	class-1	class-2	class-3	class-4	class-5	class-6
class-1	**95**	0	0	1	0	0
class-2	0	**101**	4	5	2	5
class-3	0	8	**103**	3	0	5
class-4	0	8	7	**95**	6	2
class-5	0	2	4	4	**104**	1
class-6	1	2	2	5	0	**109**

Class-1 exhibited near-perfect separation (95/96 correct). Symmetric class-2 $\leftrightarrow$ class-4 confusion (8 errors each direction) suggested vocabulary overlap. Class-3 confused with multiple categories (8 with class-2, 5 with class-6), indicating distributed similarity. Strong diagonal dominance confirmed overall accuracy.

4 Critical Reflection

4.1 Potential Improvements

Performance could be enhanced through character n-grams capturing morphological patterns, part-of-speech tag distributions revealing stylistic differences, or named entity features encoding domain-specific entities. Ensemble methods combining LinearSVC with Logistic Regression (88% dev accuracy) could improve robustness via diverse decision boundaries. Deep learning approaches (BERT, CNNs) would likely achieve higher accuracy through contextual embeddings, though requiring substantially greater computational resources. Class imbalance techniques (SMOTE, focal loss) might specifically improve underperforming categories like class-4.

4.2 Potential Biases

The model exhibits significant **temporal bias** due to dated source data. Vocabulary drift renders period-specific terms ("modem", "BBS") outdated while modern terminology ("cloud", "API") remains absent. Also, **Demographic bias** exists as training data reflects predominantly Western, English-speaking, technically literate users, limiting generalisation to diverse populations or non-standard dialects. **Topic bias** risks correlations where keyword presence ("JPEG" $\rightarrow$ graphics) overrides semantic understanding, potentially misclassifying contextually different usage (medical imaging). **Domain overfitting** to newsgroup categorisation could likely limits transferability to other text classification tasks.

4.3 Ethical Considerations

Bias increase as a systematic disadvantage for non-standard dialect speakers if standard English training data dominates. Privacy violations could occur through unauthorised monitoring or user profiling without consent. Responsible model implementation requires human oversight for major decisions, transparency mechanisms for automated decisions, and explicit consent frameworks in line with GDPR.

5 Conclusion

This pipeline achieved 88.74% test accuracy with 0.8906 macro-F1. Development experiments validated optimal feature selection ($k = 15,000$) and model choice (LinearSVC). While classical ML techniques proved effective, critical analysis identified improvement avenues (deep learning, ensembles) and inherent biases (temporal, demographic, domain). Future works could and should be explored on pre-trained language models and conduct comprehensive fairness evaluations before production deployment.