INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Ingeniería en Sistemas Computacionales

Text classification

Práctica 3

NATURAL LANGUAGE PROCESSING

Integrantes:

García Quiroz Gustavo Ivan

Hernández Medina Ulises

Reyes Núñez Sebastián

Saucedo Moreno César Enrique

Profesor:

Juárez Gambino Joel Omar.

Grupo 7CV2

30/04/2025

2025 ~ 1

Index

1 Task to be Solved

This project involves implementing a text classification system using machine learning techniques to categorize academic papers from the arxiv_normalized_corpus.csv dataset. The task requires predicting the research section or category of papers based on their textual content.

The classification problem specifically involves:

- Using the concatenated "Title" and "Abstract" columns as features for the classification models
- Predicting the "Section" column as the target class (which includes categories such as "Computation and Language" and "Computer Vision and Pattern Recognition")
- Implementing and comparing different text representation methods
- Evaluating the performance of various machine learning algorithms

The corpus was split into training (80%) and testing (20%) sets using scikit-learn's train_test_split function with shuffling enabled and a random seed (random_state=0) to ensure reproducibility of results.

2 Selected Machine Learning Methods

For this text classification task, we used and evaluated four different machine learning algorithms:

1. **Naive Bayes (MultinomialNB)**: A probabilistic classifier based on Bayes' theorem that is particularly suitable for document classification problems. Naive Bayes assumes feature independence, which works well with the bag-of-words representations used in text classification.
2. **Logistic Regression**: A linear model for binary classification that estimates probabilities using a logistic function. Despite its simplicity, logistic regression often performs well for text classification tasks, especially when the number of features is large compared to the number of observations.
3. **Support Vector Machine (SVC)**: A powerful discriminative classifier that finds an optimal hyperplane to separate data points of different classes. For this implementation, we used a linear kernel which is typically effective for text classification with high-dimensional feature spaces.
4. **Multi-layer Perceptron (MLP)**: A class of feedforward artificial neural network that can learn non-linear decision boundaries. MLPs can capture complex patterns in text data that simpler linear models might miss.

Each algorithm was tested with three different text representation methods (binary, frequency, and TF-IDF) to determine the optimal combination for classifying the academic papers.

3 Adjusted Hyperparameters

To optimize model performance, we adjusted several hyperparameters for each machine learning algorithm:

1. Naive Bayes (MultinomialNB):
   - Used default parameters from scikit-learn's implementation
   - No specific hyperparameter tuning was performed as the default configuration often works well for text classification
2. Logistic Regression:
   - max_iter: Set to 200 to ensure convergence while maintaining computational efficiency
   - Other parameters were kept at their default values
3. Support Vector Machine (SVC):
   - kernel: Set to 'linear' which is typically effective for text classification tasks
   - C: Set to 1.0 (the regularization parameter) to balance the trade-off between achieving a low training error and a low testing error
   - Other parameters were kept at their default values
4. Multi-layer Perceptron (MLP):
   - hidden_layer_sizes: Set to (100,) to provide sufficient model capacity while avoiding overfitting
   - max_iter: Set to 300 to ensure convergence of the optimization algorithm
   - Other parameters were kept at their default values (activation='relu', solver='adam', etc.)

For text representation, we explored three different vectorization approaches:

- **Binary**: Converting text into binary vectors indicating word presence/absence
- **Frequency**: Using raw word count frequencies
- **TF-IDF**: Term Frequency-Inverse Document Frequency weighting to emphasize important words

All text vectorization methods used unigrams (ngram_range=(1,1)) as specified in the requirements, focusing on individual words as features rather than phrases or n-grams.

4 Classification Report of Each Experiment

Our experiments evaluated a total of 12 model configurations, combining 4 different machine learning algorithms with 3 text representation methods. Below, we present detailed classification reports for each experiment, analyzing their precision, recall, F1-score, and overall performance metrics.

4.1 Naive Bayes with Frequency Representation

Naive Bayes with Frequency Representation

Class	Precision	Recall	F1-score	Support
Computation and Language	0.88	0.9	0.89	31
Computer Vision and Pattern Recognition	0.89	0.86	0.88	29
Accuracy			0.88	60
Macro Avg	0.88	0.88	0.88	60
Weighted Avg	0.88	0.88	0.88	60

The Naive Bayes model with frequency representation achieved an F1-score macro of 0.8830. This model demonstrated balanced performance across both classes, with slightly higher precision for "Computer Vision and Pattern Recognition" (0.89) but better recall for "Computation and Language" (0.90).

4.2 Naive Bayes with Binary Representation

Naive Bayes with Binary Representation

Class	Precision	Recall	F1-score	Support
Computation and Language	0.9	0.9	0.9	31
Computer Vision and Pattern Recognition	0.9	0.9	0.9	29
Accuracy			0.9	60
Macro Avg	0.9	0.9	0.9	60
Weighted Avg	0.9	0.9	0.9	60

Using binary representation improved the Naive Bayes model's performance, achieving an F1-score macro of 0.8999. The model showed perfectly balanced precision and recall (0.90) across both classes, suggesting that word presence/absence is more informative than frequency for this classification task.

4.3 Naive Bayes with TF-IDF Representation

Naive Bayes with TF-IDF Representation

Class	Precision	Recall	F1-score	Support
Computation and Language	0.9	0.9	0.9	31
Computer Vision and Pattern Recognition	0.9	0.9	0.9	29
Accuracy			0.9	60
Macro Avg	0.9	0.9	0.9	60
Weighted Avg	0.9	0.9	0.9	60

The Naive Bayes model with TF-IDF representation matched the performance of the binary representation, with an F1-score macro of 0.8999. This suggests that for Naive Bayes, both binary and TF-IDF representations are equally effective for this particular classification task.

4.4 Logistic Regression with Frequency Representation

Logistic Regression with Frequency Representation				
Class	Precision	Recall	F1-score	Support
Computation and Language	0.77	0.97	0.86	31
Computer Vision and Pattern Recognition	0.95	0.69	0.8	29
Accuracy			0.83	60
Macro Avg	0.86	0.83	0.83	60
Weighted Avg	0.86	0.83	0.83	60

Logistic Regression with frequency representation achieved an F1-score macro of 0.8286, the lowest among all tested configurations. This model showed an imbalance between classes, with high recall (0.97) but low precision (0.77) for "Computation and Language" and the opposite pattern for "Computer Vision and Pattern Recognition" (precision 0.95, recall 0.69).

4.5 Logistic Regression with Binary Representation

Logistic Regression with Binary Representation				
Class	Precision	Recall	F1-score	Support
Computation and Language	0.88	0.94	0.91	31
Computer Vision and Pattern Recognition	0.93	0.86	0.89	29
Accuracy			0.9	60
Macro Avg	0.9	0.9	0.9	60
Weighted Avg	0.9	0.9	0.9	60

Binary representation significantly improved Logistic Regression's performance to an F1-score macro of 0.8996. The model showed good balance between precision and recall, with slightly higher recall for "Computation and Language" (0.94) and higher precision for "Computer Vision and Pattern Recognition" (0.93).

4.6 Logistic Regression with TF-IDF Representation

Logistic Regression with TF-IDF Representation				
Class	Precision	Recall	F1-score	Support
Computation and Language	0.86	0.97	0.91	31
Computer Vision and Pattern Recognition	0.96	0.83	0.89	29
Accuracy			0.9	60
Macro Avg	0.91	0.9	0.9	60
Weighted Avg	0.91	0.9	0.9	60

The Logistic Regression model with TF-IDF representation achieved an F1-score macro of 0.8990, slightly lower than the binary representation. This model showed a similar pattern to the frequency representation but with better overall performance, having high recall (0.97) for "Computation and Language" and high precision (0.96) for "Computer Vision and Pattern Recognition".

4.7 SVC with Frequency Representation

SVC with Frequency Representation				
Class	Precision	Recall	F1-score	Support
Computation and Language	0.88	0.9	0.89	31
Computer Vision and Pattern Recognition	0.89	0.86	0.88	29
Accuracy			0.88	60
Macro Avg	0.88	0.88	0.88	60
Weighted Avg	0.88	0.88	0.88	60

The SVC model with frequency representation achieved an F1-score macro of 0.8830, showing balanced performance across both classes similar to the Naive Bayes model with the same representation.

4.8 SVC with Binary Representation

SVC with Binary Representation				
Class	Precision	Recall	F1-score	Support
Computation and Language	0.9	0.9	0.9	31
Computer Vision and Pattern Recognition	0.9	0.9	0.9	29
Accuracy			0.9	60
Macro Avg	0.9	0.9	0.9	60
Weighted Avg	0.9	0.9	0.9	60

Binary representation improved the SVC model's performance to an F1-score macro of 0.8999. Like Naive Bayes with binary representation, this model achieved perfectly balanced precision and recall (0.90) across both classes.

4.9 SVC with TF-IDF Representation

SVC with TF-IDF Representation				
Class	Precision	Recall	F1-score	Support
Computation and Language	0.9	0.9	0.9	31
Computer Vision and Pattern Recognition	0.9	0.9	0.9	29
Accuracy			0.9	60
Macro Avg	0.9	0.9	0.9	60
Weighted Avg	0.9	0.9	0.9	60

The SVC model with TF-IDF representation matched the performance of its binary counterpart, with an F1-score macro of 0.8999. This consistent performance across binary and TF-IDF representations was observed across multiple algorithms.

4.10 MLP with Frequency Representation

MLP with Frequency Representation				
Class	Precision	Recall	F1-score	Support
Computation and Language	0.88	0.9	0.89	31
Computer Vision and Pattern Recognition	0.89	0.86	0.88	29
Accuracy			0.88	60
Macro Avg	0.88	0.88	0.88	60
Weighted Avg	0.88	0.88	0.88	60

The MLP model with frequency representation achieved an F1-score macro of 0.8830, showing performance metrics identical to the SVC and Naive Bayes models with the same representation.

4.11 MLP with Binary Representation

MLP with Binary Representation				
Class	Precision	Recall	F1-score	Support
Computation and Language	0.9	0.9	0.9	31
Computer Vision and Pattern Recognition	0.9	0.9	0.9	29
Accuracy			0.9	60
Macro Avg	0.9	0.9	0.9	60
Weighted Avg	0.9	0.9	0.9	60

Binary representation improved the MLP model's performance to an F1-score macro of 0.8999, again showing perfectly balanced precision and recall (0.90) across both classes.

4.12 MLP with TF-IDF Representation

MLP with TF-IDF Representation				
Class	Precision	Recall	F1-score	Support

Computation and Language	0.9	0.9	0.9	31
Computer Vision and Pattern Recognition	0.9	0.9	0.9	29
Accuracy			0.9	60
Macro Avg	0.9	0.9	0.9	60
Weighted Avg	0.9	0.9	0.9	60

The MLP model with TF-IDF representation maintained the same performance as its binary counterpart, with an F1-score macro of 0.8999.

5 Best configuration

Machine Learning Method	**ML Method Parameters**	**Vectorizer Type**	**N-gram range**	**Average F1-score (macro)**	**N-gram range**	**Average F1-score (macro)**
Naive Bayes	default	binary	(1,1)	0.8999	(2,2)	0.8665
Logistic Regression	max_iter=200	binary	(1,1)	0.8996	(2,2)	0.8830
SVC	kernel='linear', C=1.0	binary	(1,1)	0.8999	(2,2)	0.8665
MLP	hidden_layer_sizes=(100), max_iter=300	TF-IDF	(1,1)	0.8999	(2,2)	0.8833