B23CM1047 NLU Assignment 1 Report

1. Introduction

Text classification is a fundamental task in Natural Language Processing (NLP) where the objective is to automatically assign a category label to a given document. In real-world applications, this is widely used in news classification, spam detection, sentiment analysis, topic modeling, content filtering, and recommendation systems.

In this project, the goal is to build a machine learning classifier that reads a text document and classifies it into one of two categories:

SPORT

POLITICS

The classification task is performed using standard machine learning techniques. The report includes dataset collection, preprocessing, feature representation methods, training and evaluation of at least three machine learning models, quantitative comparisons, and a discussion of limitations.

2. Dataset Collection and Preparation

2.1 Dataset Source

For this task, the **BBC News Classification Dataset** was used. This dataset is commonly used for document classification experiments and contains news articles labeled into five categories:

business

entertainment

politics

sport

tech

In this project, only the **sport** and **politics** categories were selected to form a binary classification dataset.

2.2 Dataset Format

The dataset was provided in a CSV file (bbc_data.csv) with the following columns:

data: the text of the news article

labels: the category label of the article

2.3 Filtering for Binary Classification

Since the task is specifically SPORT vs POLITICS, the dataset was filtered such that only documents with labels:

sport

politics

were retained.

After filtering, the dataset contained:

Total documents: 928

Classes: SPORT and POLITICS

2.4 Data Splitting Strategy

A train/test split was used:

Training: 80%

Testing: 20%

To ensure fair representation of both classes in training and testing, a **stratified split** was used. This ensures that the class distribution remains approximately consistent across both sets.

3. Data Preprocessing

Text preprocessing is essential to reduce noise and standardize input text.

The preprocessing steps applied were:

1. **Lowercasing**

   All documents were converted to lowercase to avoid treating words like "Government" and "government" as separate tokens.

2. **Tokenization**

   Tokenization was handled internally by scikit-learn vectorizers.

For Bag of Words and TF-IDF, tokens are extracted using default word tokenization rules.

3. **No stemming/lemmatization**

   No stemming or lemmatization was used to keep the pipeline simple and interpretable.

4. **No stopword removal**

   Stopwords were not removed. In topic classification, even common words can sometimes carry stylistic or contextual information.

4. Feature Representation Techniques

Machine learning algorithms cannot directly operate on raw text. Therefore, text documents must be converted into numerical feature vectors.

Three feature representation techniques were used in this project:

4.1 Bag of Words (BoW)

Bag of Words represents each document as a vector of word counts.

Each dimension corresponds to a word in the vocabulary.

The value is the frequency of that word in the document.

Advantages

Simple and effective for topic classification.

Works well with Naive Bayes.

Limitations

Does not capture word order.

Common words can dominate the feature space.

4.2 TF-IDF (Term Frequency–Inverse Document Frequency)

TF-IDF improves BoW by reducing the weight of very frequent words and emphasizing words that are more discriminative.

$$
TFIDF(w,d) = TF(w,d) \times IDF(w)
$$

where:

$$
IDF(w) = \log\left(\frac{N}{df(w)}\right)
$$

(N) is the number of documents

(df(w)) is the number of documents containing word (w)

Advantages

Helps identify topic-specific words.

Often improves performance for linear models.

Limitations

Still ignores word order.

4.3 TF-IDF with n-grams (Unigrams + Bigrams)

To include limited word-order information, bigrams were added:

Unigrams: single words

Bigrams: sequences of two consecutive words

This captures patterns such as:

“prime minister”

“general election”

“world cup”

“match winner”

Advantages

Captures short context and phrases.

Often improves performance.

Limitations

Increases feature space significantly.

5. Machine Learning Models Used

At least three machine learning techniques were implemented and compared.

5.1 Multinomial Naive Bayes (MNB)

Multinomial Naive Bayes is widely used for text classification because it models word counts effectively.

It assumes conditional independence between words:

[
P(y|x) \propto P(y)\prod_{i=1}^{n} P(x_i|y)
]

Why it works well

Works naturally with BoW.

Fast and effective on topic-based datasets.

5.2 Logistic Regression (LR)

Logistic Regression is a linear classifier that predicts probability of a class:

[
P(y=1|x) = \sigma(w^Tx + b)
]

where (\sigma) is the sigmoid function.

Why it works well

Strong baseline for text classification.

Handles TF-IDF well.

Produces interpretable weights.

5.3 Linear Support Vector Machine (Linear SVM)

Linear SVM finds a hyperplane that maximizes the margin between classes.

$$
\min \frac{1}{2}||w||^2 \quad \text{subject to} \quad y_i(w^Tx_i+b)\ge 1
$$

Why it works well

Often one of the best methods for sparse text classification.

Works extremely well with TF-IDF and n-grams.

6. Experimental Setup

6.1 Training and Testing

Train/Test split: 80/20

Random seed: 42

Evaluation metrics: Accuracy, Precision, Recall, F1-score

6.2 Models Compared

Feature Representation	Model
Bag of Words	Multinomial Naive Bayes
TF-IDF	Logistic Regression
TF-IDF (1,2) n-grams	Linear SVM

7. Results and Quantitative Comparison

The models produced the following results on the test set:

7.1 Bag of Words + Multinomial Naive Bayes

Accuracy: 1.00

Precision/Recall/F1 for both classes: 1.00

This indicates that the BoW + Naive Bayes model perfectly separated SPORT and POLITICS in this split.

7.2 TF-IDF + Logistic Regression

Accuracy: 0.989

POLITICS recall slightly lower (0.98), SPORT recall = 1.00

This model performed extremely well, but a small number of POLITICS articles were misclassified.

7.3 TF-IDF (Unigrams + Bigrams) + Linear SVM

Accuracy: 0.989

Similar performance to Logistic Regression.

This suggests that the dataset is already highly separable using unigram features, and bigrams provide limited additional benefit in this particular binary task.

8. Discussion of Results

8.1 Why is accuracy so high?

Achieving near-perfect accuracy is realistic in this task because SPORT and POLITICS articles contain very distinct vocabulary.

For example:

SPORT words

match, goal, striker, coach, tournament, league, team, player

POLITICS words

government, parliament, election, minister, policy, senate, bill

These words strongly signal the topic category, making the classification problem easier compared to tasks such as sentiment analysis.

8.2 Model comparison insights

Naive Bayes performed best in this experiment due to the strong topic separation.

Logistic Regression and SVM also performed extremely well.

The addition of bigrams did not significantly improve performance, suggesting unigram features are already sufficient.

9. Limitations of the System

Despite high accuracy, the system has several limitations:

9.1 Limited domain

The classifier is trained only on BBC articles. If the input text is from another source (e.g., social media, blogs), performance may drop.

9.2 Binary classification only

The system classifies only SPORT vs POLITICS. It does not handle other categories such as business or tech.

9.3 Vocabulary dependence

If an article avoids typical keywords (e.g., a sports article written without sports terms), classification may fail.

10. Conclusion

This project successfully implemented a document classification system to distinguish SPORT and POLITICS articles using machine learning. Three feature representations and three machine learning models were evaluated. The results show that classical ML methods perform extremely well on topic classification tasks, especially when classes are strongly separable.

The system achieved:

100% accuracy using Bag of Words + Naive Bayes

~99% accuracy using TF-IDF + Logistic Regression

~99% accuracy using TF-IDF n-grams + Linear SVM

The results demonstrate that traditional machine learning remains highly effective for structured text classification problems, especially when combined with appropriate feature representations.