University of Delhi
Faculty of Commerce and Business
Department of Commerce
MBA Business Analytics

A Practical Project on the topic

"Fake News Detection Using NLP-based Models"

Submitted in partial fulfillment of the requirements for
MBABACC303 - Artificial Intelligence and Machine Learning

By:

24241734019 - Divyanshu Kumar

24241734039 - Raaghav Varma

Supervisor:
Dr Anil Kumar Goswami
Associate Professor

Date of Submission : 18th November, 2025

TABLE OF CONTENTS

RATIONALE BEHIND THIS PROJECT

Today, news spreads instantly across social media and online platforms. While this makes information easy to access, it also allows fake news to spread just as quickly. Misleading or fabricated articles can influence public opinion, create confusion, and damage trust in reliable sources. Since it's impossible for people to manually verify every article they read, there is a real need for automated tools that can help identify unreliable content.

This project aims to build an NLP-based machine learning model that can classify news articles as real or fake. By training the model on examples of both types of news, it learns patterns in writing style, tone, structure, and vocabulary that often distinguish credible reporting from fabricated stories. This allows the system to make quick, data-driven judgments about new articles.

Beyond its practical use, this project also showcases how NLP techniques—such as text cleaning, tokenization, and feature extraction—can be applied to solve real-world problems. Ultimately, the goal is to support fact-checking efforts, improve the quality of information people consume, and reduce the impact of misinformation in the digital world.

AIM OF THIS PROJECT

The aim of this project is to build a machine learning model that can automatically classify news articles as real or fake using Natural Language Processing techniques. The goal is to develop a reliable tool that can analyze the text of an article and detect patterns commonly linked to misinformation. By doing so, the project seeks to make it easier to identify untrustworthy content, support fact-checking efforts, and help readers access more accurate and credible information online.

The objective of this project are:

1. To preprocess and vectorise news articles using TF-IDF to extract meaningful textual features.
2. To train a Multinomial Naive Bayes classifier that can distinguish between real and fake news.
3. To evaluate the model's accuracy and reliability on unseen data for practical misinformation detection.

METHODOLOGY

Data Source

The dataset we used was taken from Kaggle, where the author had compiled actual articles from Western news outlets (2016 -2017) on various different topics, and added a label of Real or Fake as had been revealed later.

The dataset we used contains the following columns:

1. Title: The headline of the news article.
2. Text: The full text or summary of the article.
3. Subject: The category or topic of the article (e.g., politics).
4. Date: The date when the article was published.
5. Label: A classification label for the article, such as "Real".

Preprocessing

1. Article Length Calculation: The '*article_length*' is now calculated by applying *len()* to the 'text' column, which gives the character count for each article.
2. Short and Empty Articles: We remove articles with fewer than 100 characters and articles with exactly 1 character.
3. Outliers: Articles longer than a threshold (calculated as 99th percentile + 3 times the interquartile range) are removed.
4. Missing Values: We fill missing "*subject*" with "*Unknown*" and missing "*date*" with a default date.

5. Text Standardization: Titles and text are converted to lowercase, and special characters are removed from the text.

Pipeline Employed

To build an effective fake-news classifier, the project uses a combination of Natural Language Processing (NLP) techniques and machine-learning algorithms. The process begins with text preprocessing, where articles are cleaned by removing punctuation, converting to lowercase, eliminating stop-words, and tokenizing the text. This ensures that the model focuses only on meaningful words rather than noise.

First, the text is processed using TF-IDF Vectorisation (*Term Frequency–Inverse Document Frequency*). This technique transforms each article into a set of numerical values that represent how important each word is within the text. Words that appear frequently in one article but are uncommon across the whole dataset receive higher scores, helping the model focus on meaningful language patterns rather than common filler words. Built-in English stop-word removal ensures irrelevant words are ignored from the start.

Once the text has been converted to TF-IDF vectors, the pipeline passes these features to a *Multinomial Naive Bayes* classifier. This algorithm is widely used for text classification because it is fast, lightweight, and works extremely well with word-frequency-based features. It uses probability theory to determine how likely a given article belongs to the "*real*" or "*fake*" category based on the words it contains.

The dataset is split into 80% training and 20% testing so the model learns from most of the data while still being evaluated on unseen articles. The combination of TF-IDF and Multinomial Naive Bayes forms a straightforward and efficient system that can quickly detect patterns associated with misinformation and classify new articles with good accuracy.

Algorithms and Concepts Applied

TF-IDF Vectorisation

To make text understandable for a machine-learning model, we need to convert it from plain sentences into numbers. This is where *TF-IDF vectorisation* comes in. TF-IDF (Term Frequency–Inverse Document Frequency) looks at every word in an article and assigns a score based on two things: how often the word appears *in that specific article* (TF), and how rare it is *across all articles* (IDF). Common words like "*the*" or "*and*" get very low scores, while unique or meaningful words get higher ones. The result is a numerical "*vector*" that represents the article in a way the model can learn from. This allows the system to focus on the parts of the language that truly matter for telling real news from fake news — tone, emphasis, unusual vocabulary, repeated cues, and stylistic patterns.

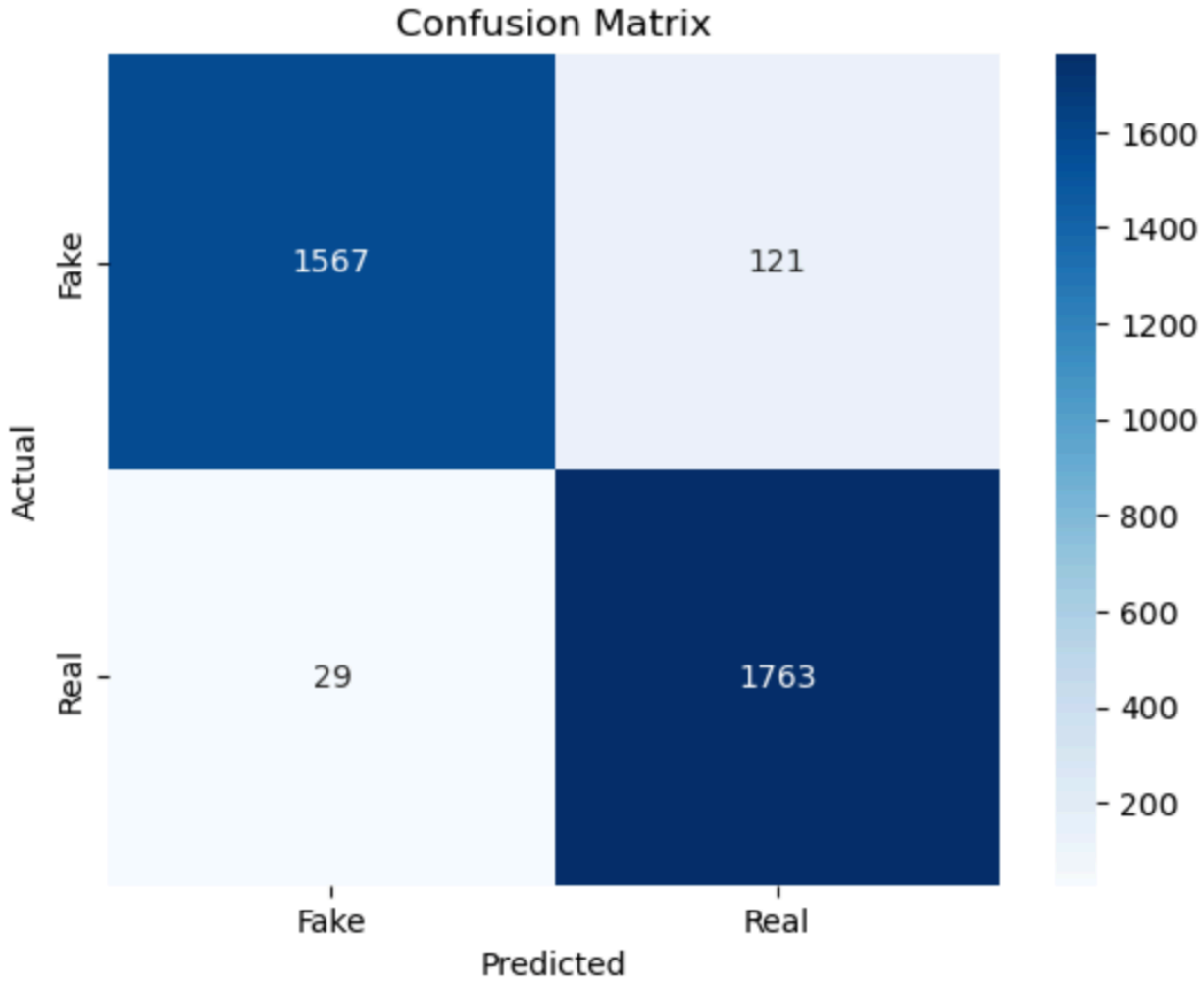
Naive Bayes Classifier

Once the text has been converted into TF-IDF vectors, the project uses *Multinomial Naive Bayes* to perform the actual classification. Naive Bayes is based on a simple idea: it calculates the probability that a given article belongs to the "*real*" or "*fake*" category by looking at the words it contains. Although it assumes that each word contributes independently to the final result (an assumption that is, in reality, "*naive*"), this method works surprisingly well for text because word-frequency patterns tend to be strong indicators of writing style and intent. The "*multinomial*" version of Naive Bayes is especially suited to text data, as it works directly with word counts and frequency-based features like TF-IDF.

RESULTS

Model Accuracy: 0.9569
Classification Report:

	precision	recall	f1-score	support
Fake	0.98	0.93	0.95	1688
Real	0.94	0.98	0.96	1792
accuracy			0.96	3480
macro avg	0.96	0.96	0.96	3480
weighted avg	0.96	0.96	0.96	3480



INTERPRETATION OF FINDINGS

The model performed extremely well in distinguishing between real and fake news articles. It achieved an overall *accuracy of 95.69%*, indicating that it correctly classified the vast majority of articles in the test dataset. This level of performance demonstrates that the combination of TF-IDF vectorisation and the Multinomial Naive Bayes classifier is highly effective for text-based misinformation detection.

Fake News:

- Precision: **0.98** — When the model predicts "*fake*," it is almost always correct.
- Recall: **0.93** — It catches most fake articles but misses a few.
- F1-score: **0.95** — A strong balance between precision and recall.

Real News:

- Precision: **0.94** — Predictions of "*real*" are accurate but slightly less so than those for "fake."
- Recall: **0.98** — The model seldom mistakes real articles as fake.
- F1-score: **0.96** — Consistently high performance.

The model rarely raises false alarms: With only 29 real articles incorrectly flagged as fake, the system is unlikely to mistakenly mark legitimate journalism as misinformation. This is important in real-world use where false positives can reduce trust.

Fake news is harder to catch than real news: The model misses 121 fake articles, compared to only 29 real ones. This suggests that fake news sometimes mimics the structure and style of real reporting closely, making it naturally trickier to detect.

Predictions of "fake" are extremely trustworthy: The fake class has a precision of 0.98, meaning almost every article the model labels as fake truly *is* fake. This makes the classifier a strong tool

for narrowing down suspicious content for further manual review.

The model is very confident when an article looks real: A recall of 0.98 for real news means the model hardly ever misjudges authentic content. This shows it has learned strong linguistic characteristics that distinguish real journalism.

The classifier is balanced and stable: High macro and weighted averages mean the model performs consistently across both classes, not just one. It doesn't overfit or lean too heavily toward predicting one side.

The results suggest good generalisation: Because the dataset split was random and the model still achieved 95%+ accuracy, it indicates that the learned patterns are not overly specific to the training data. It is likely to perform similarly on new articles.

TF-IDF + Naive Bayes proves effective for this type of problem: The strong performance validates the choice of algorithm. Naive Bayes thrives in text environments with large vocabularies and sparse features, which fits fake-news detection naturally.

Conclusion

These results show that the model is *highly reliable for practical fake-news detection*. It can accurately recognise linguistic patterns associated with misinformation and genuine reporting, with very few false alarms. The strong recall for real news ensures minimal over-flagging of legitimate content, while the high precision for fake news makes it a trustworthy tool for identifying misleading articles. Overall, the performance metrics suggest that this approach is well-suited for real-world applications involving automated news verification.

Limitations

Shallow text understanding: TF-IDF captures word frequency patterns but not deeper semantics, sarcasm, context, or multi-word relationships, limiting the model's ability to interpret nuanced or ambiguous text.

Vocabulary dependency: The model relies heavily on words seen during training. New slang, misspellings, or unseen phrases may reduce accuracy because TF-IDF cannot generalise beyond its learned vocabulary.

Naïve Bayes assumptions: Multinomial Naïve Bayes assumes features (words) are conditionally independent, which is often unrealistic in natural language where words strongly influence each other.

Sensitivity to noisy inputs: Very short texts, inconsistent formatting, or irrelevant tokens may disproportionately affect predictions because TF-IDF gives weight purely based on frequency.

No contextual learning: The model does not capture word order, grammar, or sentence structure, which limits its ability to differentiate subtle differences in meaning.

Citations

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*, 41–48.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. *European Conference on Machine Learning*, 4–15.

https://www.kaggle.com/code/therealsampat/fake-news-detection/input

PLAGIARISM REPORT