****

**Iraqi Ministry of Higher Education**

**And Scientific Research**

**Ninevah University**

**College of Information Technology**

Software Department

**“News Classification Using NLP”**

**A Project Submitted to the Council of the College of**

**Information Technology-Ninevah university**

**as a Partial Fulfillment of Requirements**

**for the Degree of Bachelor of Science**

**in**

**Software**

**BY**

**Hanan Salih**

**Raheed Fadi**

**Shahad Abdulabari**

**SUPERVISED BY**

### Dr. Ali Mohsin

### 2024-2025 A.D. 1446-1447 A.H.

**Supervisor's Declaration**

I declare that the project entitled:

**News Classification Using NLP**

was prepared by:

**Hanan Salih, Raheed Fadi and Shahad Abdulabari**

under my supervision, in partial fulfillment of the requirements for the degree of Bachelor of Science in Software Department, IT college, Ninevah University.

**Signature:**

**Supervisor Name:**

**Date:**

Acknowledgment

First and foremost, we would like to express our gratitude to God for His grace, love, and guidance, without which this project would not have been possible.

Our sincere and deep appreciation goes to our supervisor, Dr. Ali Mohsin, for his kind guidance and unwavering support. He led us on this journey, directed us towards the right path, and provided us with insightful and inspiring advice throughout the entirety of our project.

Our gratitude also extends to our families and all our friends for their support, valuable assistance, and the time we spent together.

**بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ**

}وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ{هود 11:88

**صدق الله العظيم**

} فَرِحِينَ فِي الرَّجَاءِ، صَابِرِينَ فِي الضِّيقِ، مُوَاظِبِينَ عَلَى الصَّلَاة { رُومِيَةُ ١٢:١٢

ABSTRACT

News is information obtained from different sources such as television, internet, newspapers, and magazines. Online news is published in very large numbers, and because there are so many news sources, it becomes challenging for users to find pertinent information that matches their preferences. The objective of this project is to categorize news so that a specific category can be obtained quickly and easily. Text classification in Natural Language Processing (NLP) involves categorizing and assigning predefined labels or categories to text documents, sentences, or phrases based on their content. It aims to automatically determine the class or category to which a piece of text belongs. This task is fundamental in NLP with numerous practical applications, including sentiment analysis, spam detection, topic labeling, and language identification. Building on this fundamental task, this project investigates the use of NLP techniques for classifying news articles into predefined categories. Three machine learning algorithms—Support Vector Machine (SVM), Neural Network (NN), and Random Forest (RF)—were employed to evaluate the performance of two text vectorization techniques: Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). The model was trained on The AG news classification dataset which contains over 120,000 news articles categorized into 4 classes: World, Sports, Business or Sci/Tech, At first we split the data into two parts: 80% for training and 20% for testing, results were compared to determine which combination of algorithm and technique yields the best accuracy for news classification, and were evaluated using four key performance metrics: precision, recall, F1-score, accuracy, and computation time. TF-IDF with SVM achieved the highest classification accuracy, while BoW with RF showed the best time efficiency.

الملخص

تُعد الأخبار معلومات يتم الحصول عليها من مصادر مختلفة مثل التلفزيون، الإنترنت، الصحف، والمجلات. ومع تزايد أعداد الأخبار المنشورة على الإنترنت بشكل كبير، يصبح من الصعب على المستخدمين العثور على المعلومات المناسبة التي تتوافق مع اهتماماتهم. يهدف هذا المشروع إلى تصنيف الأخبار بحيث يمكن الوصول إلى فئة معينة بسرعة وسهولة. يتضمن تصنيف النصوص في معالجة اللغة الطبيعية (NLP) تصنيف وتحديد الفئات أو التسميات المحددة مسبقًا للمستندات أو الجمل أو العبارات النصية بناءً على محتواها. ويهدف إلى تحديد الفئة أو التصنيف الذي ينتمي إليه النص تلقائيًا. تُعد هذه المهمة أساسية في مجالNLP، ولها العديد من التطبيقات العملية مثل تحليل المشاعر، واكتشاف الرسائل المزعجة (Spam)، ووضع تسميات للمواضيع، وتحديد اللغة. استنادًا إلى هذه المهمة الأساسية، يستكشف هذا المشروع استخدام تقنيات معالجة اللغة الطبيعية لتصنيف المقالات الإخبارية ضمن فئات محددة مسبقًا. تم استخدام ثلاثة خوارزميات تعلم آلي: آلة الدعم الناقل (SVM)، الشبكات العصبية، وغابة القرار العشوائية (Random Forest)، لتقييم أداء تقنيتين لتحويل النصوص إلى تمثيل عددي: Term Frequency-Inverse Document Frequency (TF-IDF) و Bag of Words (BoW). تم تدريب النماذج على مجموعة بيانات AG news classification dataset، والتي تحتوي على أكثر من 120,000 مقال إخباري مصنف ضمن أربع فئات: العالم، الرياضة، الأعمال، والعلوم/التقنية. في البداية، تم تقسيم البيانات إلى قسمين: 80% للتدريب و20% للاختبار. ثم تمت مقارنة النتائج لتحديد أفضل توليفة من الخوارزمية والتقنية من حيث دقة التصنيف، وتم تقييم الأداء باستخدام أربعة مقاييس رئيسية: الدقة (precision)، الاستدعاء (recall)، مقياس F1، الدقة العامة (accuracy)، وزمن التنفيذ. حققت تقنية TF-IDF مع خوارزمية SVM أعلى دقة تصنيف، في حين أظهرت BoW مع خوارزمية Random Forest أفضل كفاءة زمنية.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
|  | TITLE | PAGE |

[DECLARATION ii](#_Toc188267507)

[Acknowledgment iv](#_Toc188267508)

[ABSTRACT v](#_Toc188267509)

[TABLE OF CONTENTS vii](#_Toc188267510)

[LIST OF TABLES x](#_Toc188267511)

[LIST OF FIGURES xi](#_Toc188267512)

[LIST OF ABBREVIATIONS xii](#_Toc188267513)

[CHAPTER 1 INTRODUCTION 1](#_Toc188267515)

[1.1 Overview 1](#_Toc188267516)

[1.2 Problem Background 2](#_Toc188267517)

[1.3 Problem Statement 3](#_Toc188267518)

[1.4 Objectives of Study 3](#_Toc188267519)

[1.5 Project Outline 4](#_Toc188267520)

[CHAPTER 2 LITERATURE REVIEW 5](#_Toc188267521)

[2.1 Introductin 5](#_Toc188267522)

[2.2 Related Study 5](#_Toc188267522)

[2.3 Feature Extraction Techniques 6](#_Toc188267522)

[2.3.1 Bag of Words 6](#_Toc188267522)

[2.3.2 Term Frequency -Inverse Document Frequency 8](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[2.4 Classification Algorithms 11](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[2.4.1 Support Vector Machines 11](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[2.4.2 Neural Network 12](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[2.4.3 Random Forest 13](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[2.5 Limitation 14](#_Toc188267523)

[2.6 Research Gap 15](#_Toc188267524)

[CHAPTER 3 SOFTWARE MODEL 16](#_Toc188267525)

[3.1 Introduction 16](#_Toc188267526)

[3.2 Incremental Model 17](#_Toc188267527)

[3.2.1 Benefits of the Incremental Model 17](#_Toc188267522)

[3.3 Project Development Steps 18](#_Toc188267527)

[3.3.1 Analysis and Planning 19](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.2 Designing 19](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.2.1 Use Case Diagram 19](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.2.2 Functional Requirements 21](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.2.3 Activity Diagram 23](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.2.4 Sequence Diagram 25](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.3 Development 26](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.3.1 Used Libraries 27](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.3.2 Vectorization Methods 27](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.3.3 Training Algorithms 28](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.4 Testing 29](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[3.3.5 Maintenance 30](file:///C:\Users\TOTAL%20TECH%20CO\Desktop\مشروع\New%20folder\Graduation%20Research%20%5bChapters%201,%202,%203,%204%20&amp;%205%5d%20final.docx#_Toc188267522)

[CHAPTER 4 RESULTS DISCUSSION 31](#_Toc188267528)

[4.1 Dataset 31](#_Toc188267529)

[4.2 Implementation Steps 31](#_Toc188267530)

[4.3 GUI Design 31](#_Toc188267530)

[4.4 Evaluation Measurements 33](#_Toc188267530)

[4.5 Experimental Results 34](#_Toc188267531)

[CHAPTER 5 CONCLUSION AND RECOMMENDATIONS 37](#_Toc188267532)

[5.1 Research Outcomes 37](#_Toc188267533)

[5.2 Future Works 37](#_Toc188267534)

[REFERENCES 39](#_Toc188267535)

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| TABLE NO. | TITLE | PAGE |

[Table 2.1](#_Toc523259628)  Clean data and Tokenize 8

[Table 2.2](#_Toc523259628)  Term Frequency 8

[Table 2.3](#_Toc523259628)  Inverse Document Table 9

[Table 2.4](#_Toc523259628)  final equation 9

[Table 2.5](#_Toc523259628)  IDF and TF values 10

[Table 4.1](#_Toc523259628)  Performance evaluation 33

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NO. | TITLE | PAGE |

[Figure ‎3.1 Software Engineering Layers 15](#_Toc534814909)

[Figure ‎3.2 Incremental Model](#_Toc534814910) 17

[Figure ‎3.3 Use Case Diagram 19](#_Toc534814911)

[Figure ‎3.4 Functional Requirements](#_Toc534814912) 21

[Figure ‎3.5 Activity Diagram](#_Toc534814912) 23

[Figure ‎3.6 Sequence Diagram](#_Toc534814912) 24

[Figure ‎4.1 Part of the used dataset](#_Toc534814913) 30

[Figure ‎4.2 User Interface (GUI)](#_Toc534814913) 31

[Figure ‎4.3 TF-IDF comparison results](#_Toc534814913) 35

[Figure ‎4.4 BOW comparison results](#_Toc534814914) 35

LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| NLP | - | Natural Language Processing |
| TF-IDF | - | Term Frequency-Inverse Document Frequency |
| BoW | - | Bag of Words |
| SVM | - | Support Vector Machine |
| NN | - | Neural Network |
| RF | - | Random Forest |
| MLP | - | Multilayer Perceptron |
| AI | - | Artificial Intelligence |
| FNN | - | Feedforward Neural Network |
| SDLC | - | Software Development Life Cycle |
| GUI | - | graphical user interface |
| OvO | - | One-vs-One |
| ReLU | - | Rectified Linear Unit |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# INTRODUCTION

## Overview

Often news is used for educating, entertaining and advertising too. The primary purpose of news is to keep people informed about events that are taking place in their environment that may influence them. The news covers every serious topic that every person has a right to know to live a better life. Some major news topics come under political, business, sports, national, international, technology and health categories. Typically, breaking news reach people as soon as possible.

Most of the time, breaking news is not categorized under a specific category, as people tend to prioritize it over other news. Apart from breaking news, individuals have their own preferences for news categories based on different interests. These preferences vary from person to person. People read the news for various reasons—some to stay informed about current events, others to prepare for exams, or simply as a hobby. A technology enthusiast might focus only on tech news, while a businessperson may primarily read economic sections. Therefore, the purpose of reading news differs from person to person.

News classification is an important area of research to explore various approaches and algorithms to effectively categorize news articles. Many studies in this area have examined the effectiveness of various machine learning and deep learning algorithms, such as SVM, RF, and NN [1].

In Ref [2], the authors developed an English news classification model using TF-IDF for feature extraction and applied five classifiers: Decision Tree, Random Forest, Naïve Bayes, MLP, and SVM. Among the individual models, SVM achieved the highest accuracy, demonstrating its effectiveness for news categorization tasks.

However previous study utilized TF-IDF for feature extraction and ignore other methods, which can provide appropriate way to classify news.

In this study, we employed three machine learning algorithms—SVM, NN, and RF—along with two feature extraction methods: TF-IDF and BoW. These algorithms are commonly applied in various machine learning tasks, including text classification, and were selected to assess their ability to categorize news articles into predefined categories.

The text data was pre-processed using TF-IDF and BoW to convert the articles into numerical representations, which were then used as input for the classifiers. The performance of the models was evaluated through four metrics: accuracy, precision, recall, F1-score and time to ensure reliable classification results.

## Problem Background

The NLP is a field of AI focused on enabling machines to understand, interpret, and generate human language. It bridges the gap between computers and human language, dealing with written text. In news classification, NLP is used to analyze and categorize news articles based on their content, such as labeling an article as world, sport, business, and sci/tech. Unclassified news makes chaos to the newsreaders. It's very hard to find similar content without classifying news articles, manual classification is not suitable to classify large amount of news [2]. This problem led researchers to find different ways to classify the news articles.

As the number of websites providing news on the internet has increased, it has become very difficult for the user to get the news of his own interest. As a result, it is mandatory to filter the news sentences in various categories so that users can access them easily [3].

## Problem Statement

In the digital era, the vast and continuous flow of news articles across the internet presents a significant challenge in organizing and filtering content efficiently. Manual categorization of news is impractical due to the scale, velocity, and diversity of textual data. Therefore, this project addressed three problem statements.

1. achieving high accuracy in news classification remains a challenge due to several factors, including the complexity of natural language, overlapping categories, and noise in textual content. How to identify and extract informative features that can effectively represent the semantic content of news articles for use in classification tasks.?
2. Various classification algorithms have been proposed in the literature; their performance often depends heavily on the quality of feature extraction methods and the nature of the dataset. How to use appropriate classification algorithm that can obtain accurate results?

## Objectives of Study

1. To exploit two feature extraction method: TF-IDF and BoW that can extract the meaningful information and ignore other information.
2. To classify the news using three algorithms: SVM, NN and RF.

## Project Outline

The first chapter includes overview about News Classification using NLP, problem and objectives of the study. Chapter two covers the literature review. The methodology with diagrams of designing the project are illustrated in chapter three. Chapter four involves the results and discussion. Finally, conclusions and future works are presented in chapter five.

# LITERATURE REVIEW

## Introduction

News classification is a crucial task in the field of NLP and machine learning. The ability to categorize news articles into predefined classes allows for efficient information retrieval, organization, and dissemination. With the rapid growth of digital content, researchers have explored various machine learning techniques to automate this classification process. This chapter presents a review of existing studies on news classification, focusing on different methodologies, challenges, and research gaps.

## Related Study

The author of [2] conducted a comparative study on various machine learning approaches for Bangla news classification, addressing the challenges of working with a low-resource language. They evaluated algorithms including Naive Bayes, Logistic Regression, Decision Tree, RF, and SVM on a dataset comprising Bangla news articles from categories such as politics, sports, economy, technology, and entertainment. The study found that SVM and Logistic Regression performed best in terms of accuracy and F1-score, highlighting the importance of proper text pre-processing and feature extraction techniques like TF-IDF.

Several studies have explored different machine learning techniques for news classification. A review of news classification techniques emphasized traditional text classification approaches [4]. Other works examined various text classification methods, laying the foundation for modern machine learning applications [5, 6].

Recent studies have employed supervised and unsupervised machine learning approaches for news classification. Some researchers utilized machine learning models to classify news articles, demonstrating their effectiveness in improving classification accuracy [7, 8]. Another study proposed a machine learning-based system to categorize news articles, highlighting the efficiency of these techniques [9]. Research has also applied Naïve Bayes, Support Vector Machines (SVM), and Neural Networks for Nepali news classification [10], while another study explored Naïve Bayes for Bangla news classification [11].

Deep learning and neural networks have also been utilized in recent studies. A supervised multi-label classification approach was developed for Arabic news articles, emphasizing the challenges associated with multilingual text classification [12]. Another study applied SVM for Indonesian news classification, achieving high accuracy [13]. The importance of feature engineering in improving fake news classification performance has also been demonstrated [14].

Several datasets have been employed for news classification tasks. One dataset has been widely used for benchmarking machine learning models [15]. A comparative study of English news article classification using different machine learning algorithms identified key performance differences across techniques [16]. Additionally, financial news classification has been explored for forecasting applications [17].

2.3 Feature Extraction Techniques

Feature extraction plays a vital role in text classification. Common techniques include BoW, TF-IDF and word embeddings such as Word2Vec and GloVe, in our project we used BoW and TF-IDF. These techniques transform unstructured text into structured vectors that can be fed into machine learning algorithms.

2.3.1 Bag of Words

Is a simple method and was widely used technique in NLP for representing textual data, it ignores context, word order, and grammar in instead of treating a document as a collection of words. The term "Bag of Words" conveys the notion that the approach depicts a text as an unarranged "bag" or collection of words. It is referred to as a "bag of words" since it ignores any information describing the structure or sequence of words and just considers whether or not known terms appear in the document, not where they appear [6].

The BoW method convert text into fixed-length vectors by counting how many times each word appears. (A data modeling technique). It involves two steps:

* A vocabulary of known words.
* A measure of the presence of known words.

Bow Example:

We will cover an example of two documents:

Document 1: The quick brown fox jumps over the lazy dog.

Document 2: The brown fox is quick, and the dog is lazy.

* + - 1. Preprocessing:

We’ll:

* Convert to lowercase
* Remove punctuation
* Tokenize

Document 1: quick brown fox jumps lazy dog

Document 2: brown fox quick dog lazy

* + - 1. Vocabulary Creation:

0: brown

1: dog

2: fox

3: quick

4: lazy

5: jumps

* + - 1. Vectorization: Representing each document as a vector using word frequencies:

Doc 1: [1, 1, 1, 1, 1, 1]

Doc 2: [2, 1, 1, 1, 1, 0]

2.3.2 Term Frequency - Inverse Document Frequency

Is a numerical statistic method that reflects the importance of a word in a document. It considers two primary factors: a word's frequency within a document TF and its frequency throughout the corpus of documents IDF. Generally speaking, we give each word a score that represents its significance within the document and corpus.

The TF indicates how often it occurs in a document. It is computed by taking the total number of words in a document and dividing that number by the number of times a phrase appears. The result is a value between 0 and 1.

The IDF is a measure of how important a term is across all documents. It is calculated by taking the logarithm of the total number of documents in the corpus divided by the number of documents in which the term appears. The result value is a number greater than or equal to 0. The equations below denote the TF-IDF vectorization [5].

TF = 2.1

IDF = log 2.2

TF-IDF = TF x IDF 2.3

TF-IDF Example:

We will cover an example of three documents:

Document 1:  It is going to rain today.

Document 2:  Today I am not going outside.

Document 3:  I am going to watch the season premiere.

1. Clean data and Tokenize

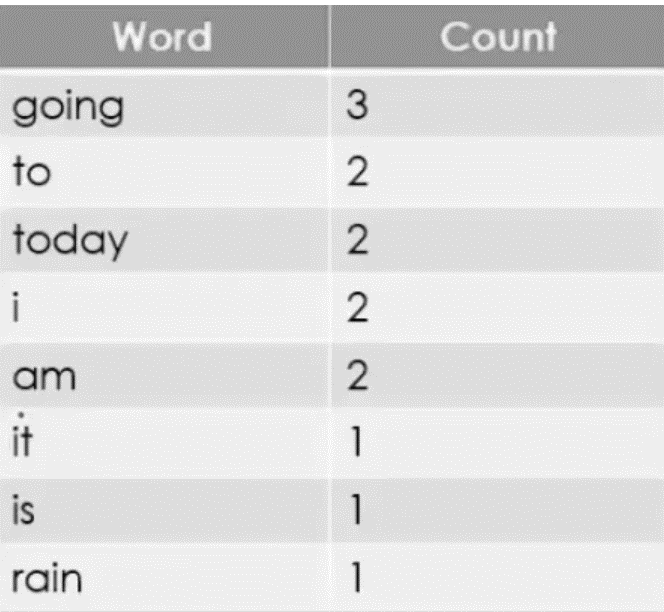


Table 2.1. Clean data and Tokenize

1. Find TF:

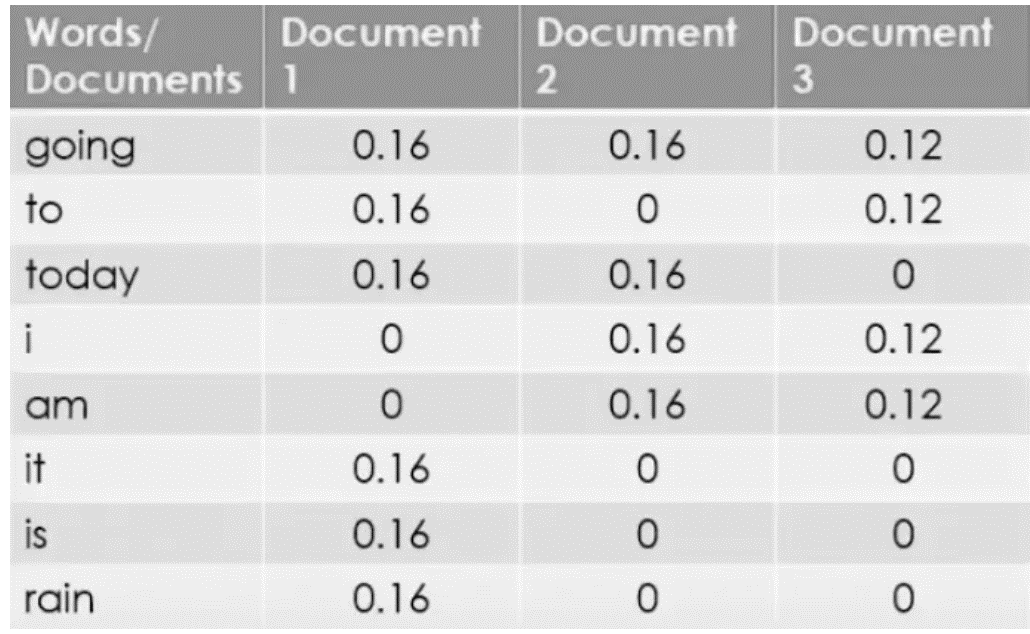


Table 2.2 Term Frequency

1. Find IDF:

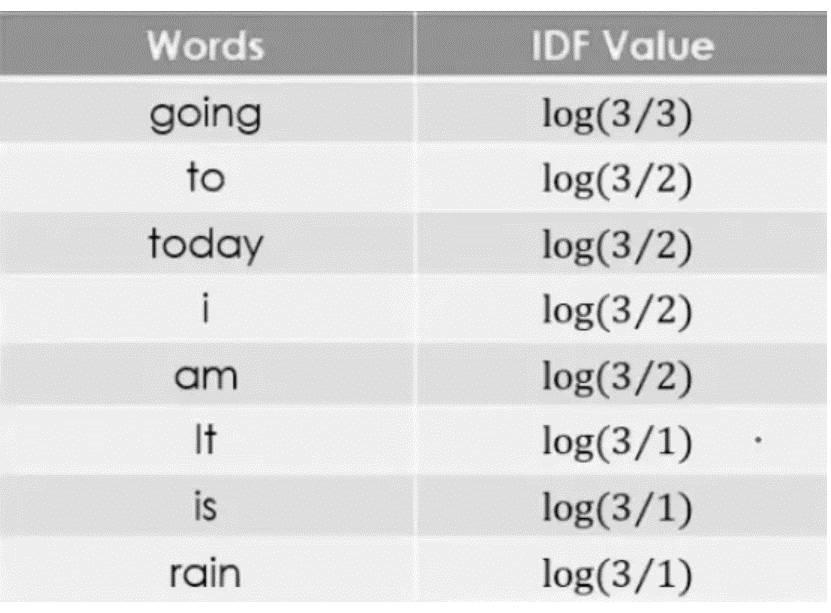


Table 2.3 Inverse Document requency

1. IDF and TF values for three documents

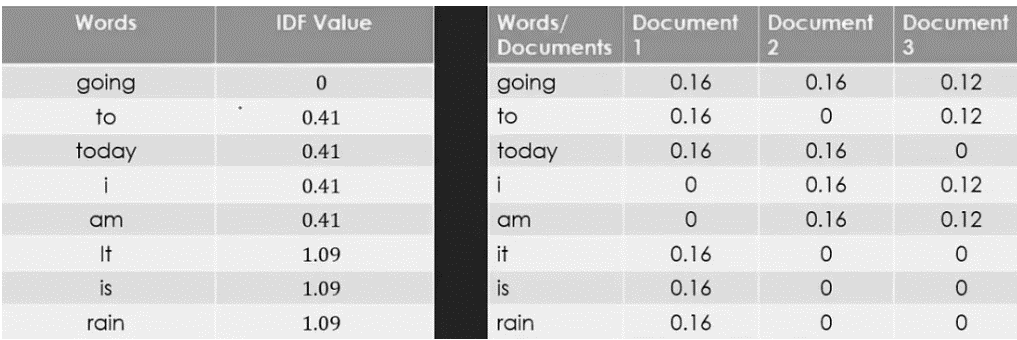


Table 2.4 IDF and TF values

1. Calculate the final equation & compare result

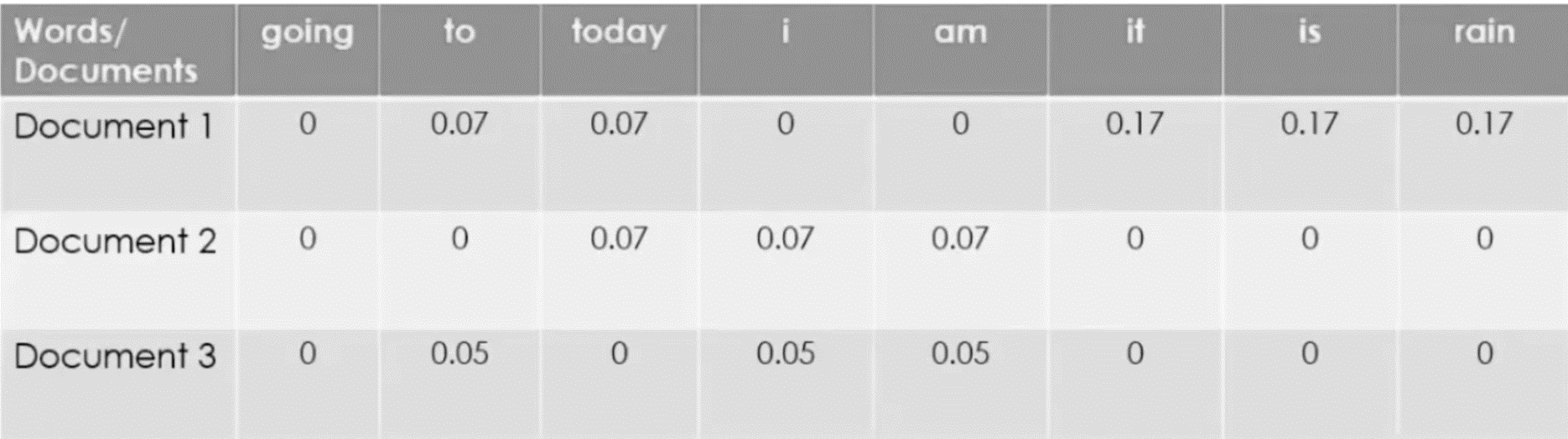


Table 2.5 final equation

2.4 Classification Algorithms

It’s one of the most fundamental tasks in NLP. It is the process of assigning predefined categories or labels to text data based on its content, involves analysing and understanding natural language (like English or Arabic sentences) to decide what class or category that text belongs to. In our project we used three training algorithms SVM, NN and RF.

2.4.1 Support Vector Machines

This technique works very well for handling multi-dimensional data, such as text representation vectors. Because it can classify text quickly and efficiently, it is also regarded as the most accurate classifier for this type of task. It works by mapping data to a high-dimensional feature space so that the data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong [10, 13].

The equation for the linear hyperplane can be written as:

2.4

Where:

* is the normal vector to the hyperplane (the direction perpendicular to it).
* is a vector containing the features of a single data point.
* is the offset or bias term, representing the distance of the hyperplane from the origin along the normal vector *w*.

To find the hyperplane that maximizes the margin:

2.5

Subject to the constraint:

2.6

Where:

* is the class label (+1 or -1) for each training instance.
* is the feature vector for the i-th training instance.
* is the total number of training instances.

The condition   ensures that each data point is correctly classified and lies outside the margin.

The distance between a data point x and the decision boundary can be calculated as:

2.7

Where:

||w|| represents the Euclidean norm of the weight vector . Euclidean norm of the normal vector .

2.4.2 Neural Network

is an algorithm that describes how the human brain processes information. It predicts and propagates a result by analyzing the values or data received in its input layer, simulating the human brain. The second layer receives the data from the input layer and forwards it to the subsequent hidden levels. The neurons or nodes in the second layer aggregate the data after identifying and filtering highly relevant patterns. The input weight is changed by assigning a weight to each input value. A logistical or sigmoid function defines and sums up these resulting values. The output of the preceding layer is examined and processed in later hidden layers before being passed on to the following layer. Then, in the output layer, values are recombined to achieve and propagate the result [7, 10, 12].

A MLP model, which is a type of FNN, was used for classification and regression tasks. The network consists of two hidden layers, containing 256 and 128 neurons, respectively.

To compute the weighted sum of the inputs:

2.8

Where, is the input feature, is the corresponding weight and  is the bias term.

The ReLU activation function was used:

2.9

To calculate the loss binary cross-entropy loss function was used:

2.10

 indicates the actual label,  refers to the predicted label, and  is the number of samples.

The Adam optimizer was used to minimize the loss function:

First moment (mean) estimate:

2.11

Second moment (variance) estimate:

2.12

and ​ are the moment estimates, ,​ are indicate rates, and represents the gradient at time

2.4.3 Random Forest

is a collection of decision trees that are trained on different subsets of the data and features. Each tree makes a prediction based on its own rules, and the final output is the average or majority vote of all the trees. Effectively, it fits a number of decision tree classifiers on various subsamples of the dataset. When creating each tree, the algorithm randomly selects a subset of features or variables to split the data rather than using all available features at a time. This adds diversity to the trees. This algorithm is currently one of best performing algorithms for many classification problems [8, 16].

Calculate Gini index to decide how nodes on a decision tree branch:

2.13

represents the number of classes, ​ is the proportion of samples of class in node .

To calculate the weighted Gini index:

2.14

​ is the number of samples in the left node, ​ refers to the number of samples in the right node, and is the total number of samples.

To calculating the feature importance:

2.15

The code's key feature is its modularity and interactivity, which make it simple to add, update, and enhance the commendation system and its interface over time.

## Limitation

Despite significant advancements, existing news classification studies face several limitations. One key challenge is the handling of multilingual news classification, as different languages present unique syntactic and semantic complexities [10, 12]. Many models trained on a single language perform poorly when applied to others, necessitating language-specific model training.

Another limitation is the difficulty of classifying fake news accurately. While content-based features have been explored for fake news detection [14], the dynamic nature of misinformation presents ongoing challenges. Additionally, classifying domain-specific news, such as financial news [17], requires specialized datasets and feature extraction techniques to ensure relevance and accuracy.

The computational cost associated with training deep learning models remains a barrier to their widespread adoption. Advanced models often require extensive computational resources, making them inaccessible for researchers with limited infrastructure [7]. Furthermore, the availability of labeled datasets is another limitation, as many studies rely on publicly available datasets that may not be representative of real-world scenarios [15].

## Research Gap

While existing studies have significantly contributed to news classification research, several gaps remain. Firstly, there is a need for more robust multilingual models that can efficiently classify news articles across different languages. Current approaches often struggle with linguistic diversity, requiring improvements in cross-lingual transfer learning and multilingual embeddings.

Secondly, the accuracy of fake news detection models needs further enhancement. Current approaches primarily rely on content-based features, but incorporating contextual, network-based, and temporal features could improve classification performance. Future research should explore hybrid models that integrate multiple feature extraction techniques.

Another research gap is the need for better domain-specific classification models. While financial news classification has been explored [17], more work is needed in other domains such as healthcare, politics, and technology. Developing customized datasets and feature extraction methods tailored to specific domains could enhance classification accuracy.

Finally, there is a need to explore lightweight and computationally efficient models for real-time news classification. Many existing studies rely on deep learning approaches that require substantial computational power. Future research should investigate methods to optimize model efficiency while maintaining high classification performance.

# SOFTWARE MODEL

## Introduction

The project's methodology is introduced in this chapter. The SDLC framework is used to produce software at every level and offers useful techniques for creating software. The technique of software engineering is multi-layered, as shown in Figure (3.1). To attain quality, any engineering approach needs to follow a certain organizational commitment [18].

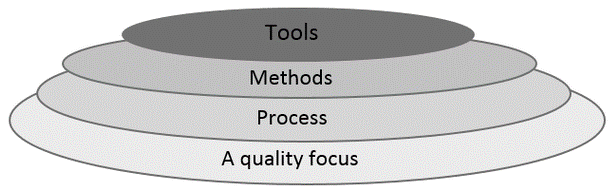


Fig 3.1 Software Engineering Layers

Project layers:

1. Quality focus: Build full functioning, free of bugs, upgradeable and maintainable system.
2. Process: Applying the SDLC and implement its stages planning, analysis, design, implementation, testing and maintenance.
3. Methods: Using an incremental approach to build the project.
4. Tools: Using number of tools to develop the project:
5. Anaconda Spyder: to write the code using python.
6. Enterprise Architect: to design the diagrams.
7. GitHub: for version control.

There are various models and diagrams used in software development and programming projects, including Waterfall Model, Incremental Model, Spiral Model, and others.

This project was developed using the Incremental Model since it has several benefits.

## Incremental Model

The incremental model combines the prototype model's iterative methodology with aspects of the sequential linear model. The Incremental Model breaks down requirements into several independent parts across a number of software development phases. Every unit passes through the phases of design, implementation, testing, and requirements. With each new iteration of the unit, the preceding version gains new capability. Using the Incremental Model, the first increase often referred to as a core product increases. The procedure keeps going until the entire system is finished.

This approach was chosen because it enables the rapid delivery of functional software early in the software lifecycle.

1. Benefits of the Incremental Model
2. More flexibility compared to other models.
3. Testing and debugging are easier during each iteration.
4. In this model, the client can respond to each build.
5. Risk management is easier because risky components are identified and dealt with during each iteration.

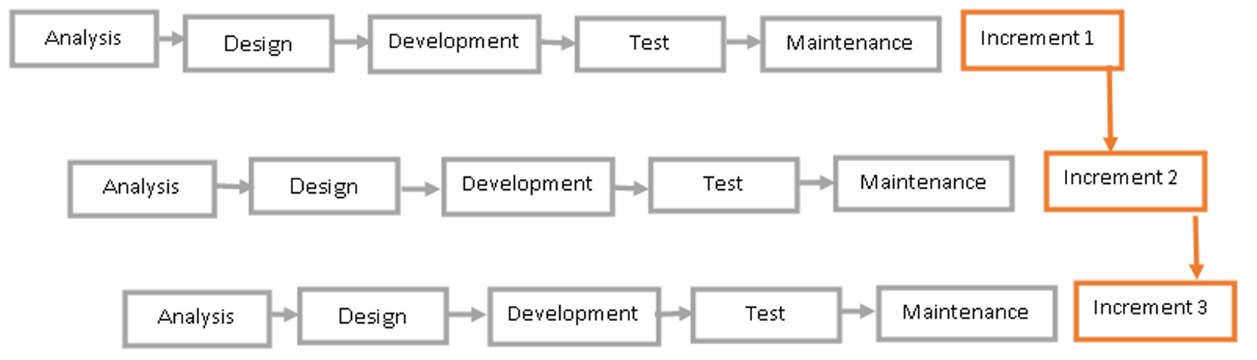


Fig 3.2 Incremental Model

In our project we followed these increments:

1. The first increment training and testing SVM algorithm using Bow method and recording the results.
2. The second increment training and testing SVM algorithm using TF-IDF method and recording the results.
3. The third increment training and testing NN algorithm using Bow method and recording the results.
4. The fourth increment training and testing NN algorithm using TF-IDF method and recording the results.
5. The fifth increment training and testing RF algorithm using Bow method and recording the results.
6. The sixth increment training and testing RF algorithm using TF-IDF method and recording the results.
7. The last increment was building the GUI

## Project Development Steps

## Planning, analysis, and other tasks are all part of the process of managing the progress of any project. The project will outline these activities in the paragraphs that follow to demonstrate this.

## Analysis and Planning

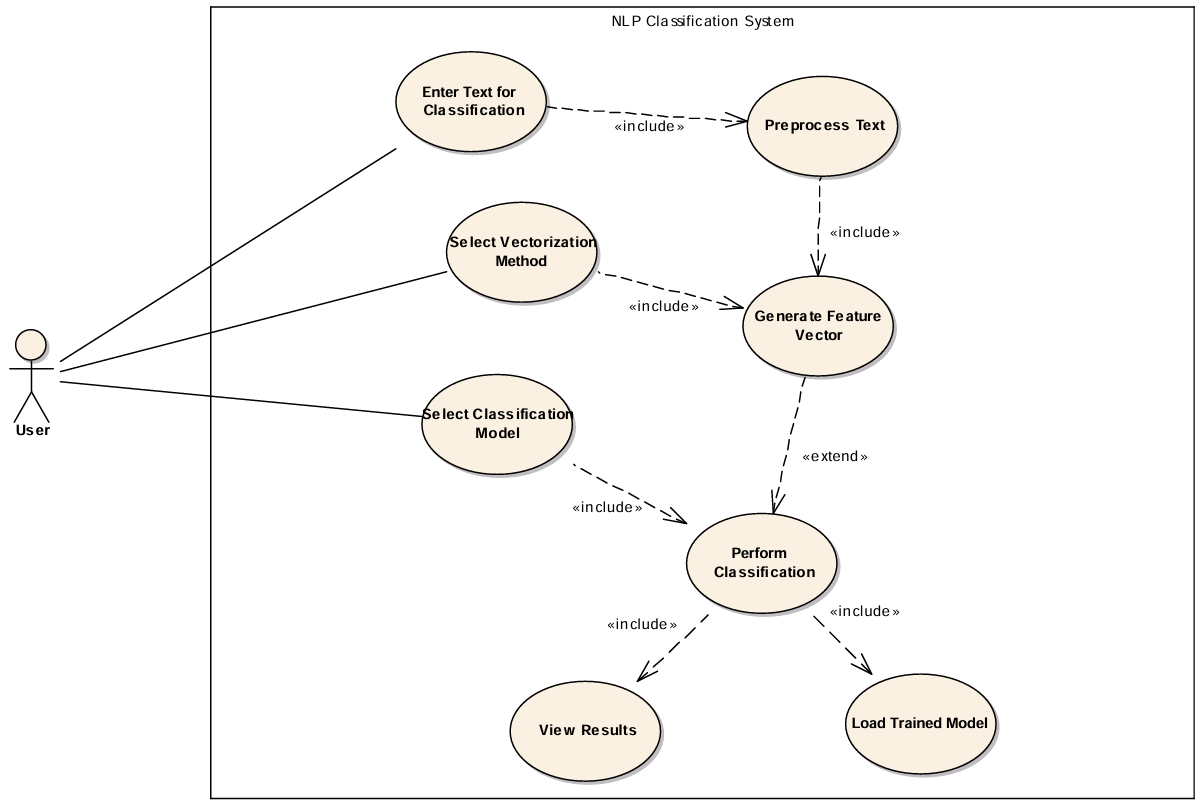
1. Understanding System Requirements: Determining the requirements and specifications of the NLP News Classification System.
2. Feasibility Study: Assessing the system development project's overall feasibility as well as its technical and financial aspects.
3. Project Planning: Identifying required resources, creating a schedule, and assigning duties and responsibilities are all part of project planning.

## Designing

1. System Design: organizing modules and components, defining their interactions, and defining the system's structure.
2. User Interface Design: Designing the GUI, including specifying interface elements and their layout.

## Use Case Diagram

In Figure (3.3) An NLP classification system's essential features are depicted in the use case diagram. Using a variety of NLP and machine learning components, it visually simulates how a user interacts with the system to classify input text.

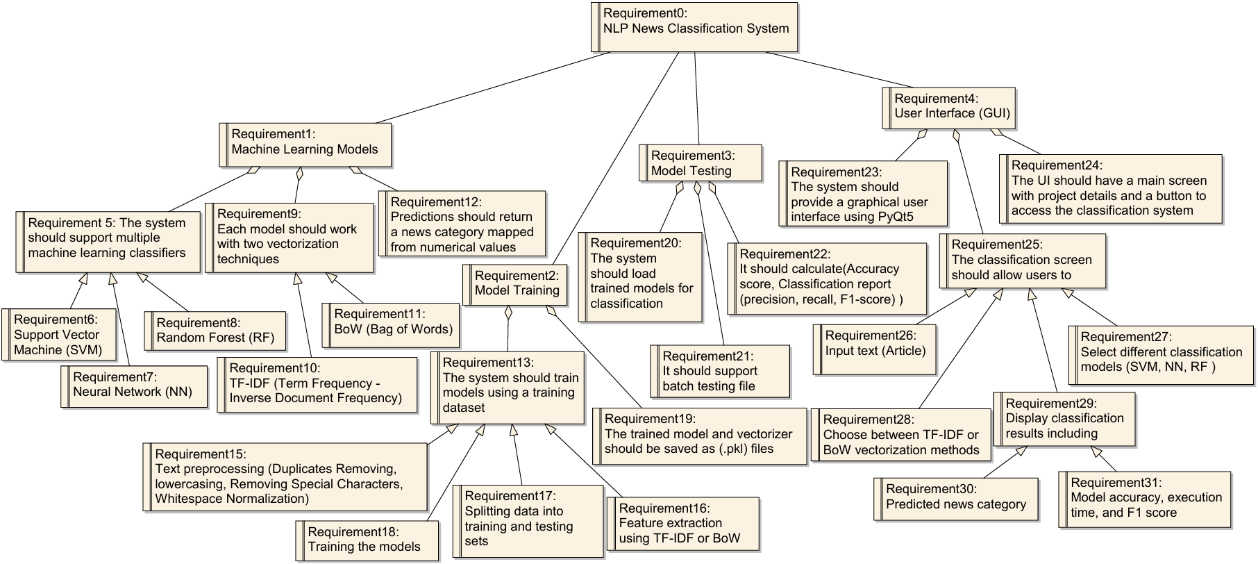


## Fig 3.3 Use Case Diagram

1. **User**: The primary actor who interacts with the system to classify input text.
2. Enter Text for Classification: The user provides input text that they want the system to classify.
3. Preprocess Text: ensures the input text is suitable for further processing by applying NLP techniques.
4. Select Vectorization Method: The user chooses a method such as BoW or TF-IDF to convert text into feature vectors.
5. Generate Feature Vector: Converts the preprocessed text into a numerical format that a machine learning model can understand.
6. Select Classification Model: The user selects a pre-trained machine learning model SVM, NN and RF to perform the classification.
7. Perform Classification: Executes the classification Algorithm using the chosen model and the generated feature vector.
8. View Results: Shows the output label or classification result to the user (predicted category, accuracy, execution, time and F1 score).
9. Load Trained Model: Retrieves a pre-trained model that will be used to classify the input text.

## Functional Requirements

In Figure (3.4) represents the structured requirements of our NLP News Classification System. It outlines different components necessary for building and deploying a machine learning-based news classification model. The system consists of various modules, including Machine Learning Models, Model Training, Model Testing, and a GUI.



## Fig 3.4 Functional Requirements

1. Machine Learning Models
   * The system should support multiple machine learning classifiers (SVM, NN and RF).
   * Each model should work with two vectorization techniques (TF-IDF and BoW).
   * Predictions should return a news category mapped from numerical values.
2. Model Training
   * The system should train models using a training dataset.
   * Training involves multiple steps (Feature extraction, splitting data into training and testing sets and training the models).
   * The trained model and vectorizer should be saved as .pkl files for reuse.
3. Model Testing

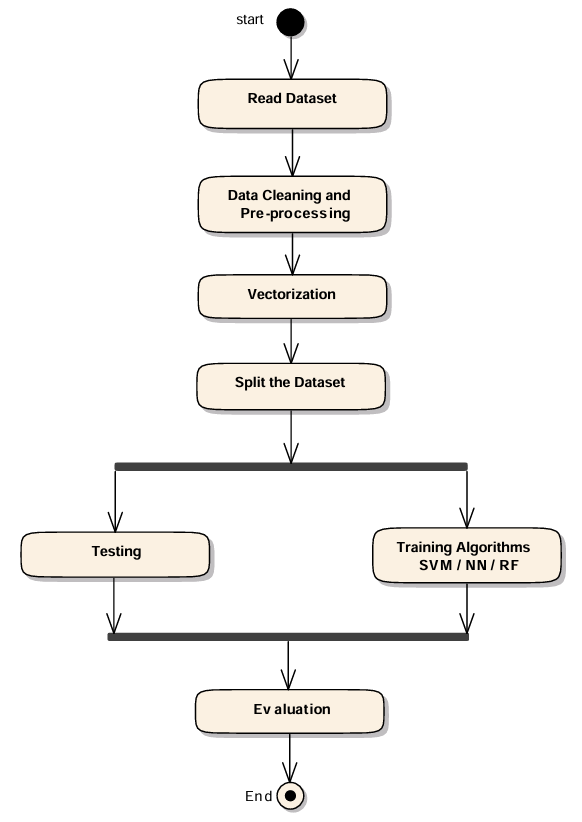
* The system should load trained models for classification.
* It should support batch testing files for efficiency.
* It should calculate performance metrics such as Accuracy, Precision, Recall, F1-score).

1. User Interface

* The system should provide a GUI using PyQt5.
* The UI should have a main screen with project details and a button to access the classification system.
* The classification screen should allow users to (Input text, select classification model, choose vectorization method and Display classification results and metrics).

## Activity Diagram

In Figure (3.5) Represents the step-by-step process used in this NLP-based news classification model. Each step plays a crucial role in preparing, training, and evaluating the classification system.

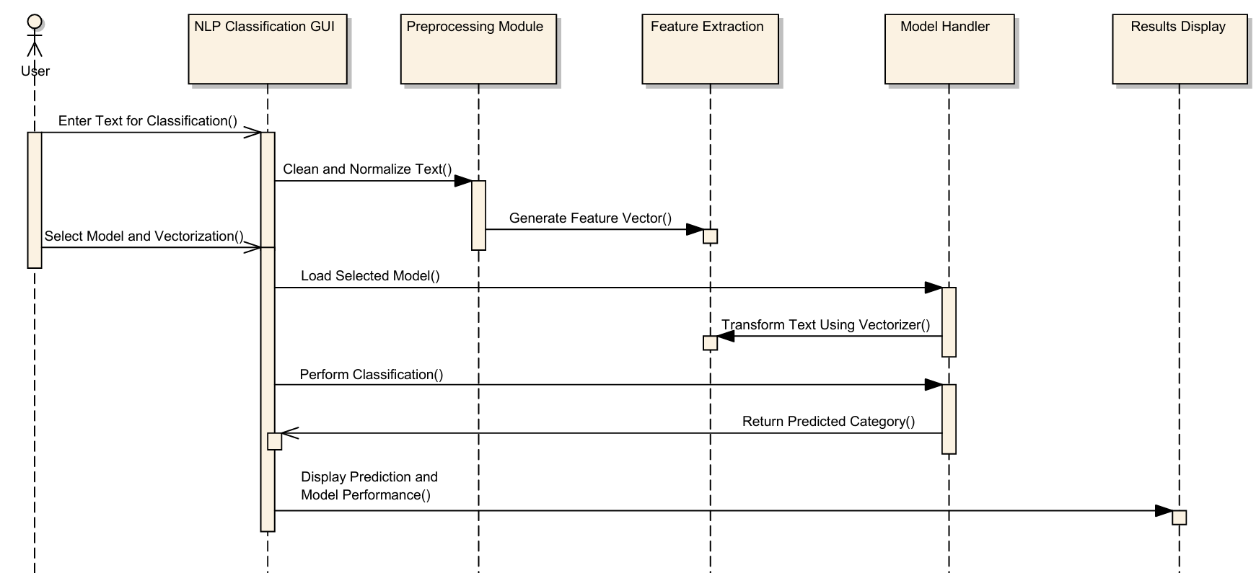


## Fig 3.5 Activity Diagram

1. Read Dataset: The process begins by loading the dataset, which consists of labeled news articles.
2. Data Cleaning and Pre-processing: cleaning and preparing the data for processing by following these steps, duplicates removal, tokenization, lowercasing, punctuation and special characters removal.
3. Vectorization: Since machine learning models cannot process raw text directly, it is transformed into numerical format using vectorization techniques (e.g. BoW and TF-IDF).
4. Split the Dataset: The dataset is divided into training and testing sets.
5. Training Algorithms: Three machine learning algorithms are applied to train the model (SVM, NN and RF).
6. Testing: After training, the model is tested using unseen data to measure its generalization performance. Various evaluation metrics.
7. Evaluation: final step involves analyzing model performance using statistical metrics such as accuracy, time, precision, recall and F1-score.

## Sequence Diagram

In Figure (3.6) represents the workflow of this NLP-based text classification system. It visually demonstrates the interaction between different system components, from user input to classification and result display.



## Fig 3.6 Sequence Diagram

1. Actors and Components

* User: The end-user who inputs text for classification.
* NLP Classification GUI: The graphical interface where the user interacts with the system.
* Preprocessing Module: Cleans and normalizes the input text to improve classification accuracy.
* Feature Extraction: Converts the preprocessed text into numerical feature vectors for the model.
* Model Handler: Loads the selected classification model and performs text classification.
* Results Display: Shows the predicted category and model performance metrics.

1. Process Flow

* User Input: The user enters text into the system for classification.
* Preprocessing: The system cleans and normalizes the text (e.g., duplicates removal, tokenization, lowercasing, punctuation, special characters removal).
* Feature Extraction: The processed text is converted into a numerical representation using a vectorization technique (e.g., BoW and TF-IDF).
* Model Selection & Loading: The system loads the user-selected trained classification model.
* Text Transformation: The input text is transformed using the chosen vectorization method.
* Classification: The selected model processes the vectorized input and predicts the category.
* Result Display: The system presents the classification result along with model performance metrics (accuracy, time, precision, recall and F1-score).

## Development

The NLP News Classification System was implemented using Python programming language and by using several libraries. Here's a breakdown of what the code does and the libraries used:

1. Importing Dataset: The code reads the dataset from a .CSV file that contains classified news articles.
2. Data Cleaning and Pre-processing: The code cleans the data for vectorization by removing duplicates, tokenization, lowercasing, punctuation and removing special characters.
3. Vectorization: The code transforms all the text into numerical values using two methods BoW and TF-IDF.
4. Split the Dataset: The code divides the dataset into two parts training and testing.
5. Training the models: The code Produces six trained models from three algorithms (SVM, NN and RF) and two vectorization methods each algorithm will be trained twice once with BoW then with TF-IDF.
6. Testing: The code uses unseen articles to test the model capabilities and gives measurements of its performance, the metrics are accuracy, time, precision, recall and F1-score.
7. GUI: PyQt5 library is used to create user friendly interface for interacting with the system. User can enter text (article) to be classified and choose the vectorization method and the algorithm to classify the text. The program shows the predicted class for the chosen algorithm and the model performance and metrics.
8. Used Libraries
9. Pandas: Used to import the dataset.
10. Re: Used to clean the text.
11. Sklearn: Used to vectorize the text, split the dataset into training, testing and to train the model and performance metrics calculation.
12. Joblib: Used to save the trained model as .pkl file.
13. Sys: Used to handle program exiting.
14. PyQt5: Used to build the GUI.
15. Vectorization Methods
16. BoW is a basic NLP method for representing text as a collection of words. It disregards grammar, word order, and context, focusing only on word presence or frequency. Each document is converted into a vector based on word counts from a predefined vocabulary.

BoW treats text like an unordered "bag" of words, ignoring their position in the text. It is simple and effective, especially for tasks like text classification and document comparison.

1. TF-IDF is a statistical method that evaluates the importance of a word in a document relative to a corpus. The TF measures how often a word appears in a document, normalized by document length. The IDF reduces the weight of common words by considering how many documents contain the term.

TF-IDF is the product of TF and IDF, giving higher scores to important, less frequent words. It’s widely used in text classification, search engines, and information retrieval tasks.

1. Training Algorithms
2. SVM: is a supervised learning algorithm widely used in text classification due to its ability to handle high-dimensional data effectively. it works by finding the optimal hyperplane that separates classes with the maximum margin. It is particularly effective in binary classification tasks and has been extended to multi-class problems through one-vs-one or one-vs-rest strategies. Studies have shown that SVM achieves high accuracy in news classification, especially when combined with appropriate feature extraction methods. Since our dataset consists of 4 classes and SVM is primarily designed for binary classification, we employed the OvO approach. The OvO method is a binary classifier is constructed for each potential pair of classes. This method is particularly advantageous when the number of classes is relatively small, as it allows for a focused comparison between pairs of classes. Each classifier is responsible for distinguishing between two specific classes, effectively ignoring the others.
3. NN: particularly deep learning models like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), have gained popularity in news classification. These models are capable of capturing complex patterns and semantic relationships within the text. Neural networks require large amounts of training data but can significantly outperform traditional algorithms when properly trained. Their ability to learn from raw text data makes them powerful tools for language modelling and classification tasks.
4. RF: is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes of the individual trees. It is known for its robustness, ability to handle overfitting, and relatively low computational cost compared to deep learning models. In news classification, Random Forest has been used to achieve competitive accuracy, particularly in cases where the dataset is noisy or contains irrelevant features. Its interpretability and ease of implementation make it a popular choice for initial model development.

## Testing

1. Unit Testing: Testing each unit separately to ensure that it functions correctly. This involves testing individual components and functions of the NLP News Classification System, such as data preprocessing, vectorization and models implementation. Unit testing ensures that each unit performs its intended function accurately.
2. Integration Testing: Testing the integration of components together to ensure proper interaction. In the context of the NLP News Classification System, integration testing involves testing how different modules and components interact with each other. This includes testing the integration between data preprocessing, vectorization, classification, models implementation and prediction. Integration testing confirms that the system works together as one unit.
3. System Testing: Testing the entire system to verify compliance with user requirements and desired quality. System testing for the NLP News Classification System involves evaluating the end-to-end functionality of the system, including data input, processing, classification, and predicted output. It ensures that the system meets the requirements, such as accuracy, efficiency, and usability. Additionally, we will split the data into 80% for training the models, and the remaining 20% will be used for testing its performance.

For evaluating the performance of the NLP News Classification System, we will use accuracy, precision, recall and F1-score. The ratio of accurately categorized articles to all articles utilized for classification is known as the accuracy rate. The model's recall, sometimes referred to as its true positive rate, is its capacity to locate every pertinent instance within a dataset. Precision is the percentage of pertinent cases that are truly pertinent. Maximizing precision comes at the expense of recall, and vice versa. In order to get around this restriction, the F1 measure was implemented.

## Maintenance

Based on the code provided, a number of tasks can be identified in the maintenance phase of an incremental model to guarantee the NLP News Classification System's continued performance, functionality, and usability. These are a few potential tasks:

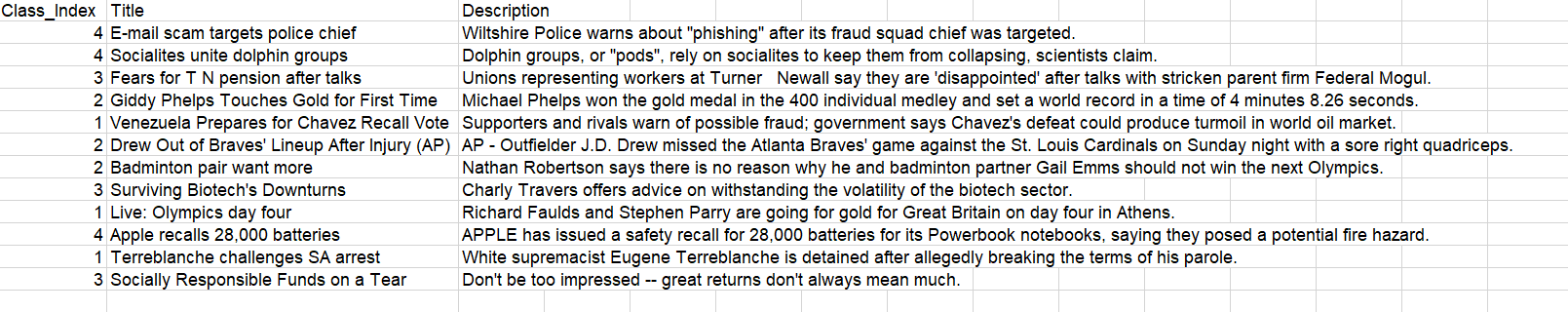
* Bug Fixing and Issue Resolution: Continuously monitor the system for any errors, bugs, or issues reported by users or detected through testing. Debug and fix any identified issues promptly to ensure smooth operation of the system.
* Performance Optimization: Analyzing and optimizing the performance of the recommendation algorithms and system functionalities to enhance efficiency and responsiveness.
* Data Updates and Maintenance: Regularly update the news dataset used by the system to include new articles.
* Algorithm Refinement and Enhancement: Explore new techniques or models to improve the quality and diversity of news classification.

# RESULTS DISCUSSION

## Dataset

AG News Classification Dataset is utilized in this research. It involves 4 classes: World, Sports, Business or Sci/Tech. Dataset has 120,000 records they're divided into two parts 80% for training consists of 96,000 samples and 20% for testing consists of 24,000 samples, evenly distributes across the four classes.

Part of the used dataset:



## Fig 4.1 Part of the used dataset

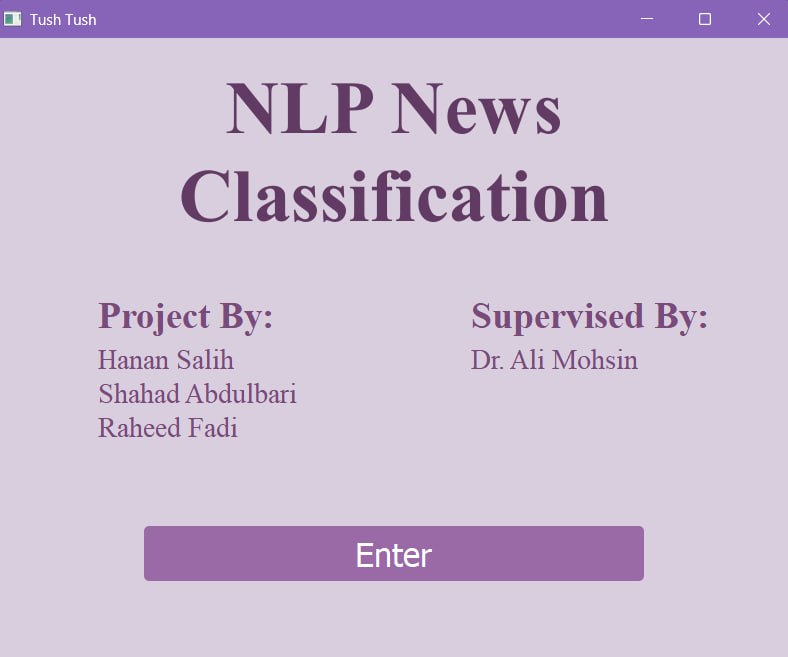
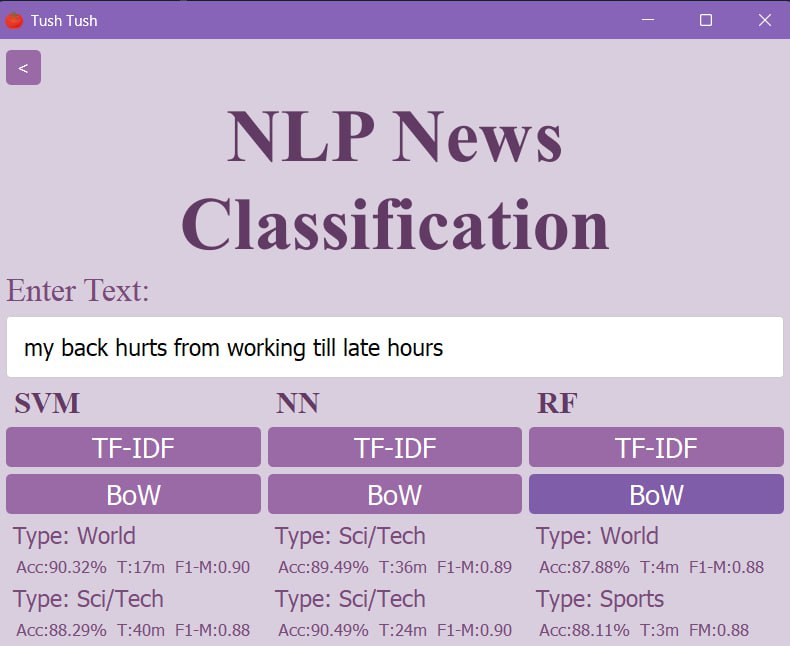
## Implementation Steps

The implementation process consisted of the following steps:

1. Data Collection: Collecting data from AG News dataset using Pandas library.
2. Data Preprocessing: Clean text, tokenize and convert to lowercase Using Re library.
3. Feature Extraction: Convert text into numerical representations, applying BoW once and then TF-IDF using Sklearn library.
4. Model Selection & Training: Three models: SVM, RF, and NN were used to train on the dataset using Sklearn library.
5. Model Evaluation: Assess performance using accuracy, precision, recall, and F1-score metrics using Sklearn library.
6. GUI Development: Designed and implemented a GUI in Python to interact with the model using PyQt5 and Sys library.

## GUI Design

Below are screenshots of the GUI developed for the project, showcasing how users interact with the model.



## Fig 4.2 GUI

The first screen displays the project's title, along with the names of the researchers and the supervisor. An "Enter" button allows the user to proceed to the second screen, where they can compare the three algorithms.

On the second screen, there is a text field where the user can enter the title of their article. They can then click on the algorithm and method buttons to classify the article into one of four categories: World, Sports, Business, or Sci-Tech, and view the accuracy of the selected algorithm.

## Evaluation Measurements

The evaluation is based on 5 metrics, are the precision, recall, F1-measure, accuracy and the time to calculate the efficiency of the algorithms that are utilized in this research. The time represents the timing for each algorithm to finish its training. The ratio of accurately categorized articles to all articles utilized for classification is known as the accuracy rate. The model's recall, sometimes referred to as its true positive rate, is its capacity to locate every pertinent instance within a dataset. It is calculated where TP is number of True Positive and FN is the number of False Negatives. In this analysis, it is calculated as News correctly classified as:

4.1

Precision is the percentage of pertinent cases that are truly pertinent. It is calculated where FP is number of False Positives. It is computed as follows:

4.2

Maximizing precision comes at the expense of recall, and vice versa. In order to get around this restriction, the F1 measure was implemented. It is computed using the following formula and represents the harmonic mean of precision and recall:

4.3

## Experimental Results

The table below provides a comprehensive comparative analysis of news classification performance. It is structured into two sections based on feature extraction techniques: BoW and TF-IDF. Each section includes three primary columns representing the classification algorithms employed. The evaluation metrics considered in this analysis are precision, recall, and F1-score, all of which are computed for each algorithm. Additionally, the average accuracy rate is derived by calculating the mean F1-score across all categories. Lastly, the training time for each algorithm is measured in minutes to assess computational efficiency.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TF-IDF | SVM | | | Neural Network | | | Random Forest | | |
| Topics | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| World | 0.92 | 0.90 | 0.91 | 0.92 | 0.90 | 0.90 | 0.90 | 0.88 | 0.89 |
| Sports | 0.95 | 0.97 | 0.96 | 0.93 | 0.95 | 0.95 | 0.90 | 0.96 | 0.93 |
| Business | 0.87 | 0.87 | 0.87 | 0.86 | 0.87 | 0.86 | 0.85 | 0.84 | 0.85 |
| Sci/Tech | 0.88 | 0.87 | 0.88 | 0.87 | 0.85 | 0.86 | 0.86 | 0.83 | 0.85 |
| Average | 0.90 | 0.90 | 0.90 | 0.89 | 0.89 | 0.89 | 0.87 | 0.87 | 0.87 |
| Average accuracy rate | | | **90.32%** |  | | 89.49% |  | | 87.88% |
| Training Time | | | **17m** |  | | 36m |  | | 4m |
|  | | | | | | | | | |
| Bow | SVM | | | Neural Network | | | Random Forest | | |
| Topics | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| World | 0.88 | 0.89 | 0.88 | 0.90 | 0.90 | 0.90 | 0.91 | 0.88 | 0.89 |
| Sports | 0.94 | 0.95 | 0.95 | 0.95 | 0.96 | 0.95 | 0.91 | 0.96 | 0.93 |
| Business | 0.85 | 0.84 | 0.85 | 0.86 | 0.86 | 0.86 | 0.84 | 0.85 | 0.84 |
| Sci/Tech | 0.86 | 0.85 | 0.85 | 0.87 | 0.86 | 0.87 | 0.86 | 0.84 | 0.85 |
| Average | 0.88 | 0.88 | 0.88 | 0.89 | 0.89 | 0.89 | 0.88 | 0.88 | 0.88 |
| Average accuracy rate | | | 88.29% |  | | 89.47% |  | | **88.11%** |
| Training Time | | | 40m |  | | 24m |  | | **3m** |

Table 4.1. Performance evaluation

For the first section TF-IDF, SVM achieves the highest average precision, recall, and F1-score (0.90), followed closely by NN (0.89), while RF lags behind (0.87). in addition, the Sports category consistently yields the highest classification scores across all models, with SVM achieving an F1-score of 0.96, the highest among all categories. On the other hand, the Business and Sci/Tech categories demonstrate slightly lower scores compared to World and Sports, indicating more challenges in distinguishing these topics effectively.

Furthermore, RF is the fastest (4 minutes), making it computationally efficient. SVM takes 17 minutes, balancing between accuracy and computational cost. NN require 36 minutes, reflecting the higher computational expense associated with deep learning models.

According to the second section, the performance ranking remains relatively similar to TF-IDF, with NN (0.89) and SVM (0.88) performing slightly better than RF (0.88) in terms of average precision, recall, and F1-score. Moreover, The Sports category continues to perform exceptionally across models, with NN achieving an F1-score of 0.95. The Business and Sci/Tech categories again report lower classification scores, reinforcing the notion that these topics may have overlapping features.

In terms of time, RF remains the fastest (3 minutes), showing efficiency in handling BoW features. NN (24 minutes) perform slightly better than with TF-IDF but still demand high computational resources. SVM takes the longest time (40 minutes), likely due to the large-dimensional feature space created by the BoW model.

It can be observed from the results, TF-IDF appears to yield slightly better classification performance, particularly with SVM and Neural Networks. While, BoW is still competitive but tends to produce slightly lower accuracy rates, likely due to its inability to capture term importance as effectively as TF-IDF.

Hence, SVM performs best in terms of accuracy and robustness, making it a suitable choice when classification performance is the priority. Whereas, NN provide comparable accuracy but at a higher computational cost, making them less practical for real-time applications unless resources are abundant. Eventually, RF is the most computationally efficient, providing decent classification performance with the lowest training time. The figures 3.1 and 3.2 demonstrate similar aspect as follows.

Fig.4.3 TF-IDF comparison results

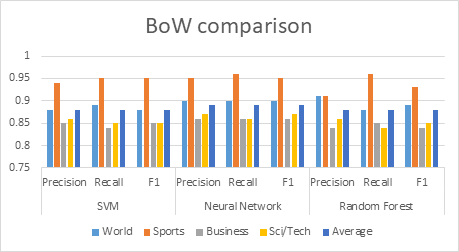


Fig. 4.4 BoW comparison results

In terms of the average accuracy rate and timing when we used TF-IDF, SVM algorithm was the best with 90.32% accuracy rate and 17m Training Time while RF was the worst with 87.88% accuracy rate and 4m Training Time, and when we used BoW NN algorithm was the best with 89.47% accuracy rate and 24m Training Time while SVM was the worst with 88.29% accuracy rate and 40m Training Time.

# CONCLUSION AND RECOMMENDATIONS

## Research Outcomes

The study categorizes news articles into four distinct classes using three classification algorithms: SVM, NN, and RF. The findings suggest that for real-world news classification applications, SVM with TF-IDF is the optimal choice when accuracy is the primary concern. Conversely, RF with BoW is more suitable for scenarios requiring faster processing with moderate accuracy. While NN demonstrate strong classification performance, they demand substantial computational resources and extended training time, making them more appropriate for large-scale, high-resource environments. Furthermore, when accuracy is the top priority, SVM with TF-IDF remains the preferred approach. If computational efficiency is a critical factor, RF with BoW serves as the best alternative. However, if deep learning solutions are viable, NN with TF-IDF offer robust classification capabilities. For future research, combining TF-IDF and BoW could be explored to leverage the advantages of both feature extraction methods. Additionally, a hybrid classification approach could be implemented to further enhance performance. Finally, incorporating an additional dataset would enable a more precise evaluation of model accuracy.

## Future Works

There are a few ways we could expand and improve our NLP News Classification System in the future. One area of enhancing could be using dataset with more categories and articles. This would involve improving the system’s ability to handle this amount of data.

Another area for future work could be incorporating additional features to enhance the system's functionality. For example, the system could be trained to recognize fake news this will improve system abilities and provides trusted system to categories any organization articles.

Additionally, the system could be integrated with other technologies to improve its overall functionality. For example, it could be linked with news organizations websites or large databases systems to help handling the large number of articles that are received every day.

Finally, combining TF-IDF and BOW could be explored to leverage the advantages of both feature extraction methods. A hybrid classification approach could be implemented to further enhance performance. Finally, incorporating an additional dataset would enable a more precise evaluation of model accuracy.

REFERENCES

1. Veziroğlu, M., Veziroğlu, E. and Bucak, İ.Ö., 2024. Performance Comparison between Naive Bayes and Machine Learning Algorithms for News Classification, in Bayesian Inference - Recent Trends.
2. Kamrus, S., Alam, M.K., Nabi, M.A., Fahim, A. & Faisal, B.A., 2021. A comparative study of different text classification approaches for Bangla news classification. 2021 24th International Conference on Computer and Information Technology (ICCIT), pp.1-6.
3. Kaur, G. and Bajaj, I.K., 2016. ISSN : 2394-4714 Foundation of Computer Science FCS, Communications on Applied Electronics (CAE).
4. Kaur, G. & Bajaj, K., 2016. News classification and its techniques: a review. IOSR Journal of Computer Engineering (IOSR-JCE), 18, pp.22-26.
5. Ikonomakis, M., Kotsiantis, S. & Tampakas, V., 2005. Text classification using machine learning techniques. WSEAS Transactions on Computers, 4, pp.966-974.
6. Korde, V. & Mahender, C.N., 2012. Text classification and classifiers: a survey. International Journal of Artificial Intelligence & Applications, 3, p.85.
7. Agarwal, J., Christa, S., Pai H, A., Kumar, M.A. & G.P.M.S, 2023. Machine learning application for news text classification. 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, pp.463-466. DOI: 10.1109/Confluence56041.2023.10048856.
8. Kamrus, S., Alam, M.K., Nabi, M.A., Fahim, A. & Faisal, B.A., 2021. A comparative study of different text classification approaches for Bangla news classification. *2021 24th International Conference on Computer and Information Technology (ICCIT)*, pp.1-6.
9. Singh, Y.V., Naithani, P., Ansari, P. & Agnihotri, P., 2021. News classification system using machine learning approach. 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, pp.186-188. DOI: 10.1109/ICAC3N53548.2021.9725409.
10. Shahi, T.B. & Pant, A.K., 2018. Nepali news classification using Naïve Bayes, support vector machines and neural networks. 2018 International Conference on Communication Information and Computing Technology (ICCICT), pp.1-5.
11. Chy, A.N., Seddiqui, M.H. & Das, S., 2014. Bangla news classification using naive Bayes classifier. 16th International Conference on Computer and Information Technology (ICCIT), pp.366-371.
12. Shehab, M.A., Badarneh, O., Al-Ayyoub, M. & Jararweh, Y., 2016. A supervised approach for multi-label classification of Arabic news articles. 2016 7th International Conference on Computer Science and Information Technology (CSIT), pp.1-6.
13. Liliana, D.Y., Hardianto, A. & Ridok, M., 2011. Indonesian news classification using support vector machine. World Academy of Science, Engineering and Technology, 57, pp.767-770.
14. Chun-Ming, L., Mei-Hua, C., Endah, K., Vinod, K.V. & Chao-Tung, Y., 2022. Fake news classification based on content level features. Applied Sciences, 12(3), pp.2-5.
15. Aman, A.E., 2020. AG News Classification Dataset. Available at: <https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset> [Accessed 24 October 2024].
16. Disayiram, N. & Rupasingha, R.A.H.M., 2021. A comparative study of classifying English news articles using machine learning algorithms. Trends in Electrical, Electronics, Computer Engineering Conference (TEECCON), pp.50-55.
17. Savaş, Y., Dhanya, J., Can, K. & Ayşe, B., 2018. Classification of "hot news" for financial forecast using NLP techniques. 2018 IEEE International Conference on Big Data (Big Data), pp.4719-4722.
18. Pressman, R.S., 2010. *Software Engineering A Practitioner’s Approach Seventh Ed*