A Minor project review on

"CLASIFICATION OF ONLINE NEWS ARTICLES USING MACHINE LEARNING MODELS"

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

in

CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

by

M. NIKITHA	22211A6667
P. VARSHITH	22211A6691
S. SHASHANK	22211A66A6
G. SUJITH	22211A66B1
T.P. VISHNU VARDHAN	22211A66B4

Under the guidance of

Supervisor : Ms. B PRIYANKA, &

Co-Supervisor : Mrs. B LAVANYA

DEPARTMENT OF
CSE (ARTIFICIAL INTELLIGENCE&MACHINE LEARNING)
B.V. RAJU INSTITUTE OF TECHNOLOGY
(UGC Autonomous) Vishnupur, Narsapur, Medak (Dist.) 502313(TS)
(Affiliated to JNTU and approved by A.I.C.T.E

2022-2026

CANDIDATE'S DECLARATION

We here by certify that the work which is being presented in the Minor Project entitled "CLASIFICATION OF ONLINE NEWS ARTICLES USING MACHINE LEARNING" in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology and submitted in the Department of CSE(Artificial Intelligence & Machine Learning), B.V.Raju Institute of Technology, Narsapur is an authentic record of our own work carried out during a period from March 2024 to June 2024 under the guidance of Ms.B Priyanka & B Lavanya. The work presented in this paper has not been submitted by us for the award of any other degree of this or any other Institute/University.

BY:

M. Nikitha 22211A6667

P. Varshith 22211A6691

S. Shashank 22211A66A6

G. Sujith 22211A66B1

T.P. Vishnu 22211A66B4

B. V. Raju Institute of Technology
(UGC Autonomous, Accredited By NBA & NAAC
Vishnupur, Narsapur, Medak ((Dist.),
Telangana State, India-502313

CERTIFICATE

This is to certify that the Minor Project entitled **"CLASIFICATION OF ONLINE NEWS ARTICLES USING MACHINE LEARNING",** being submitted by,

M. Nikitha 22211A6667
P. Varshith 22211A6691
S. Shashank 22211A66A6
G. Sujith 22211A66B1
T.P. Vishnu 22211A66B4

in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING) to** the Jawaharlal Nehru Technological University, Hyderabad is a bonafide work carried out by team under the guidance of

Ms. B Priyanka & B Lavanya, Department of CSE(AI&ML).

This is to certify that the above statement made by the students is/are correct to the best of my knowledge.

SUPERVISOR **: Ms. B Priyanka** CO-SUPERVISOR : **Ms. Lavanya**

SIGNATURE : SIGNATURE :

The Minor Project Viva-Voce for this team has been held on _____

External Examiner

Dr. G. Uday Kiran
Program Coordinator

ACKNOWLEDGEMENT

The success and final outcome of this term paper required a lot of guidance and assistance from many people, and we are extremely fortunate to have got this all along the completion. Whatever we have done is due to such guidance and assistance. We would not forget to thank them.

We thank Ms. B Priyanka, for guiding us and providing all the support in completing this term paper. We are thankful to Ms. B Priyanka Mam & B Lavanya Mam for supporting us in doing this project.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all the staff members of CSE (AI&ML) Department

B. V. Raju Institute of Technology
(UGC Autonomous, Accredited By NBA & NAAC
Vishnupur, Narsapur, Medak ((Dist.),
Telangana State, India-502313

ABSTRACT

In the era of vast digital information, the automatic classification of online news articles is crucial for efficient content organization and personalized user experiences. This research focuses on developing and optimizing machine learning models for the accurate categorization of news articles into predefined topics. The process begins with the collection and preprocessing of a diverse dataset of online news articles. Various machine learning models, including Naive Bayes, Support Vector Machines, Random Forests, and Neural Networks, are explored and optimized through rigorous training and evaluation processes. Hyperparameter tuning and feature engineering are employed to enhance model performance. The optimized models demonstrate superior accuracy, precision, recall, and F1-score metrics. The deployment of the final model enables real-time classification of unseen news articles, contributing to the efficient organization and retrieval of information in the ever-expanding digital news landscape. The outcomes of this research offer insights into the effective utilization of machine learning for online news classification and provide a foundation for future advancements in the field.

Team Members:
M. Nikitha
P.Varshith
S.Shashank
G. Sujith
Tp . Vishnu

SUPERVISOR : **Ms. B Priyanka**
SIGNATURE :

CO–SUPERVISOR : **Mrs. B Lavanya**
SIGNATURE :

INDEX:

LIST OF CONTENTS **PAGE**

LIST OF FIGURES :

CHAPTER-1
INTRODUCTION

1.INTRODUCTION :

In today's digital age, the proliferation of online news has created an unprecedented need for effective and efficient ways to manage and categorize vast amounts of information. With millions of news articles published daily across various platforms, users are often overwhelmed by the sheer volume of content. This situation necessitates robust methods to classify news articles accurately and quickly, enabling users to find relevant information effortlessly and enhancing their overall experience.

1.1.General Introduction :

Machine Learning (ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence (AI).

Machine learning (ML) models have emerged as powerful tools to address this challenge. By leveraging large datasets and advanced algorithms, ML models can learn to categorize news articles based on their content, significantly improving the organization and retrieval of news information. This process not only aids in personalized content delivery but also enhances search engines' accuracy and efficiency, contributing to better information dissemination.

The classification of online news articles involves several steps, including data collection, preprocessing, feature extraction, model training, and evaluation. Various machine learning techniques, such as Naive Bayes, Support Vector Machines (SVM), Random Forests, and neural networks, can be employed to build and optimize these models. Each technique has its strengths and weaknesses, making it crucial to experiment with different models and fine-tune them to achieve optimal performance.

1.2 Motivation for The Project:

In the rapidly evolving digital landscape, the amount of information available online is growing exponentially. News articles are a significant part of this information overload, with countless pieces of content being published every second across various platforms. This vast sea of information can be overwhelming for users who seek relevant, timely, and accurate news. Therefore, there is an urgent need for innovative solutions to manage and categorize this content effectively.

1.3 Importance of machine learning:

Consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” on Amazon—are all examples of applied machine learning. All these examples echo the vital role machine learning has begun to take in today’s data-rich world. Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide variety of industries. With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that’s in part due to the increased sophistication of machine learning, which helps analyze those big chunks of big data. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques

1.4 Uses of machine learning:

Earlier in this article, we mentioned some applications

of machine learning. To understand the concept of machine learning better, let's consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data Traditionally, data analysis was always being characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing 3 By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis.

1.5Problem Statement:

To develop and implement an efficient machine learning system for classifying online news articles into relevant categories, utilizing optimization techniques to enhance accuracy and to give personalized user experience. However, it is not possible to monitor the everyday prices of the houses. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for house price prediction.

1.6 PYTHON:

The basic programming language used for machine learning is: PYTHON.

INTRODUCTION TO PYTHON:

● Python is a high-level, interpreted, interactive, and object-oriented scripting language.

● Python is a general-purpose programming language that is often applied in scripting roles

● Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.

● Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs

● Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

1.6.1 History of python:

● Python was developed by GUIDO VAN ROSSUM in the early 1990s

● Its latest version is 3.7, it is generally called python3.7

1.6.2 Features of python:

Easy-to-learn: Python has few keywords, a simple structure, and a clearly defined syntax, this allows the student to pick up the language quickly.

● Easy-to-read: Python code is more clearly defined and visible to the eyes.

● Easy-to-maintain: Python's source code is fairly easy-to-maintaining.

● A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

● Databases: Python provides interfaces to all major commercial database.

1.6.3 How to set up Python

● Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

● The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

1.6.4 Installation (using python IDLE)

Installing python is generally easy, and nowadays many Linux and Mac OS distributions include a recent python.

● Download python from www.python.org

● When the download is completed, double-click the file and follow the instructions to install it

● When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.

Fig:1.6.1: python download

1.6.5 Installation (using anaconda)

Python programs are also executed using Anaconda

● Anaconda is a free open-source distribution of python for large-scale data processing, predictive analytics, and scientific computing.
●Conda is a package manager which quickly installs and manages packages
● Step 1: Open Anaconda.com/downloads in a web browser.
● Step 2: Download python 3.4 version for (32-bit graphics installer/64 -bit graphic installer)

- Step 3: select installation type (all users)
- Step 4: Select path (i.e., add anaconda to path & register anaconda as default python 3.4), next click install, and next click finish.
- Step 5: Open jupyter notebook (it opens in the default browser).

Fig 1.6.2 Anaconda download

Fig 1.6.3 Jupyter notebook

1.6.6 Python variable types:

●Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

●Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

●Python has five standard data types –

o Numbers

o Strings

o Lists

o Tuples

o Dictionary

1.7 Modules:

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU general public license (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Pandas:

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library It is fast, and it has high performance & productivity for users.

Pandas were initially developed by Wes Mc Kinney in 2008

while he was working at QAR Capital Management. He convinced the QAR to allow him to open source the Pandas. Another QAR employee, Chang She, joined as the second major contributor to the thelibraryin2012. Over time many versions of pandas have been released. The latest version of the pandas is 1.4.1.

Advantages:

- Fast and efficient for manipulating and analyzing data.
- Data from different file objects can be loaded.
- Easy handling of missing data (represented as Nan) in floating point as well as non-floating-point data
- Size mutability: columns can be inserted and deleted from Data Frame and higher dimensional objects
- Data set merging and joining.
- Flexible reshaping and pivoting of datasets
- Provides time-series functionality.

NumPy:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

A tuple of integers giving the size of the array along each dimension is known as the shape of the array. An array class in NumPy is called the ND array. Elements in NumPy arrays are accessed by using square brackets and can be initialized by using nested Python Lists. Creating a NumPy Array.

Arrays in NumPy can be created in multiple ways, with various numbers of Ranks, defining the size of the Array. Arrays can also be created with the use of various data types such as lists, tuples, etc. The type of

their resultant array is deduced from the type of the elements in the sequences.

Sklearn:

Sklearn also known as Scikit-learn. Scikit-learn is probably the most useful library for machine learning in Python. The Sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. Scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface.

***Important features of scikit-learn**:*

- Simple and efficient tools for data mining and data analysis.
- It features various classifications, regression, and clusters in algorithms including some as support vector machines, random forests, gradient boosting, k-means, etc.
- Accessible to everybody and reusable in various contexts.
- Built on the top of Num Py, SciPy, and matplotlib.
- Open source, commercially usable.

Flask:

Flask is an API of python that allows us to build up web applications and mobile applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple application.

1.8 Algorithms:

What is Machine Learning:

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

Machine Learning Methods:

Machine learning algorithms are often categorized as supervised, unsupervised and semi-supervised machine learning algorithms.

1. Supervised machine learning algorithm
2. Unsupervised machine learning algorithm
3. Semi-supervised machine learning algorithm

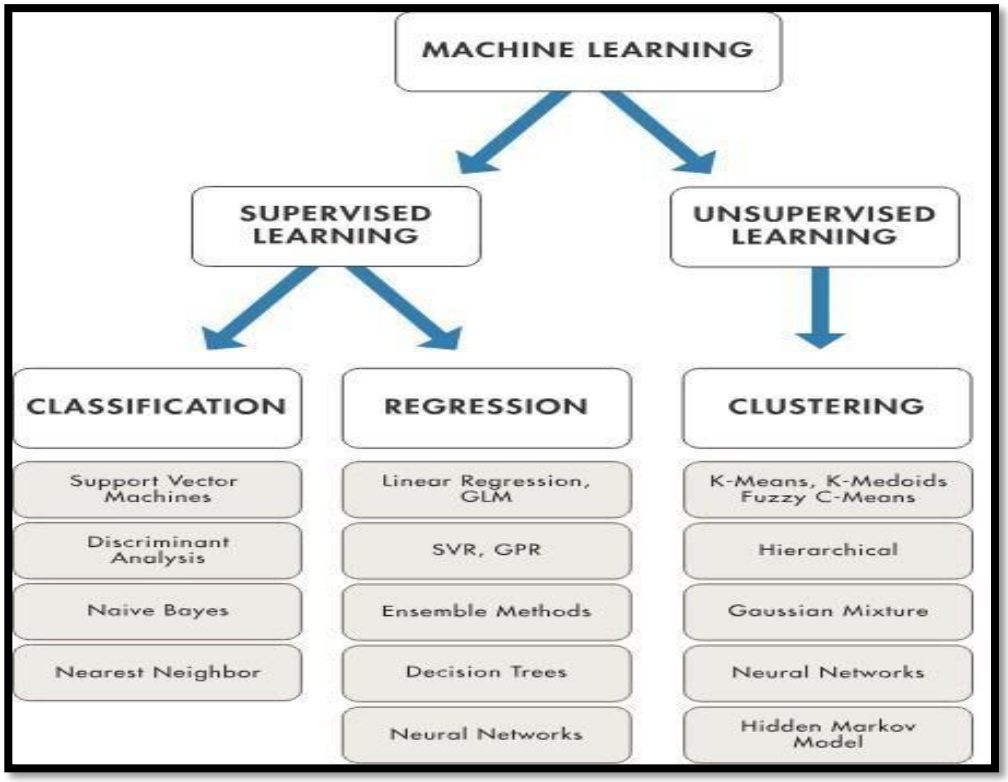

Fig 1.8.1 Machine Learning Method

1. Supervised machine learning algorithms:

A supervised machine learning algorithm can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system can provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

- Most of the practical machine learning uses supervised learning.
- Supervised learning problems can be further grouped into regression and classification problems.

1. **Classification:** Here our target variable consists of categories.
2. **Regression:** Here our target variable is continuous, and we usually try to find out the line of the curve.

Some common types of problems built on top of classification and regression include recommendation and time series prediction respectively.

There are various ways to get labeled data:

1. Historical labeled Data
2. Experiment to get data: We can perform experiments to generate labeled data like A/B Testing.
3. Crowdsourcing.

Supervised Algorithms:

The most widely used learning algorithms are:

- Support Vector Machines
- Linear regression
- naïve Bayes
- Decision trees
- K-Nearest Neighbor algorithm
- Neural Networks

2.Unsupervised machine learning algorithm:

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data. Unsupervised learning is where you only have input data (X) and no corresponding output variables. The goal of unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

These are called unsupervised learning because unlike supervised learning above there are no correct answers and there is no teacher. Algorithms are left to their own devices to discover and present the interesting structure in the data. Unsupervised learning problems can be further grouped into clustering and association problems.

1. **Clustering**: A clustering problem is where you want to discover the in here not groupings in the data, such as grouping customers by purchasing behavior.
2. **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Un supervised Algorithms:

The most widely used learning algorithms are:

- K-means
- Clustering
- Mixture models
- Hierarchical clustering
- Neural Networks
- Hebbian Learning

- Self-organizing map

Applications:

- Bioinformatics
- Database marketing
- Handwriting recognition
- Information extraction and retrieval
- Object recognition in computer vision
- Pattern recognition

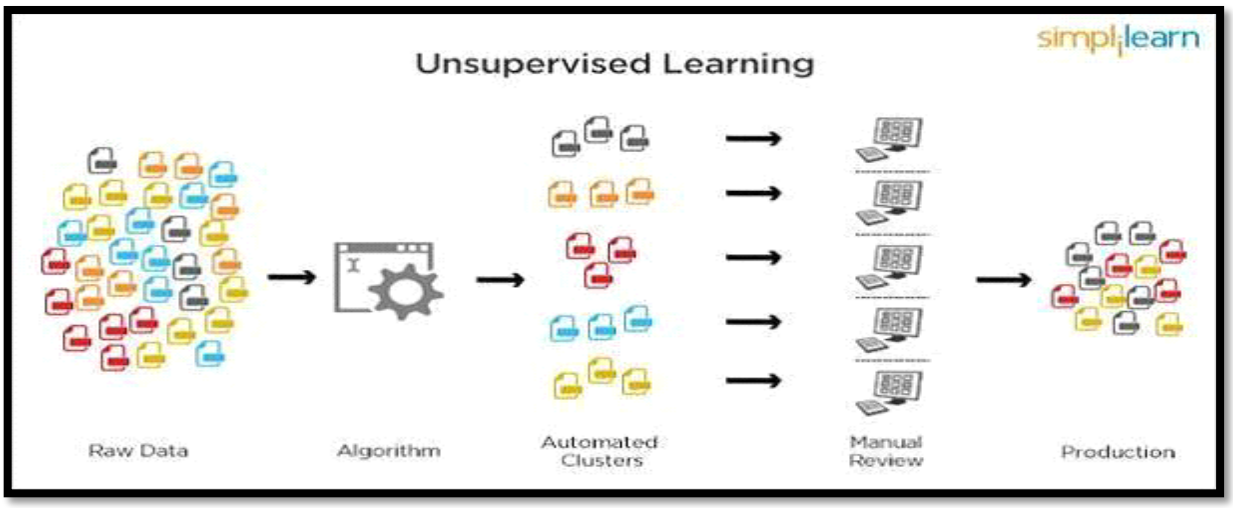

Fig 1.8.2 unsupervised learning

3. Semi-supervised learning algorithms:

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labeled and unlabeled data for training. In a typical scenario, the algorithm would use a small amount of labeled data with a large amount of unlabeled data.

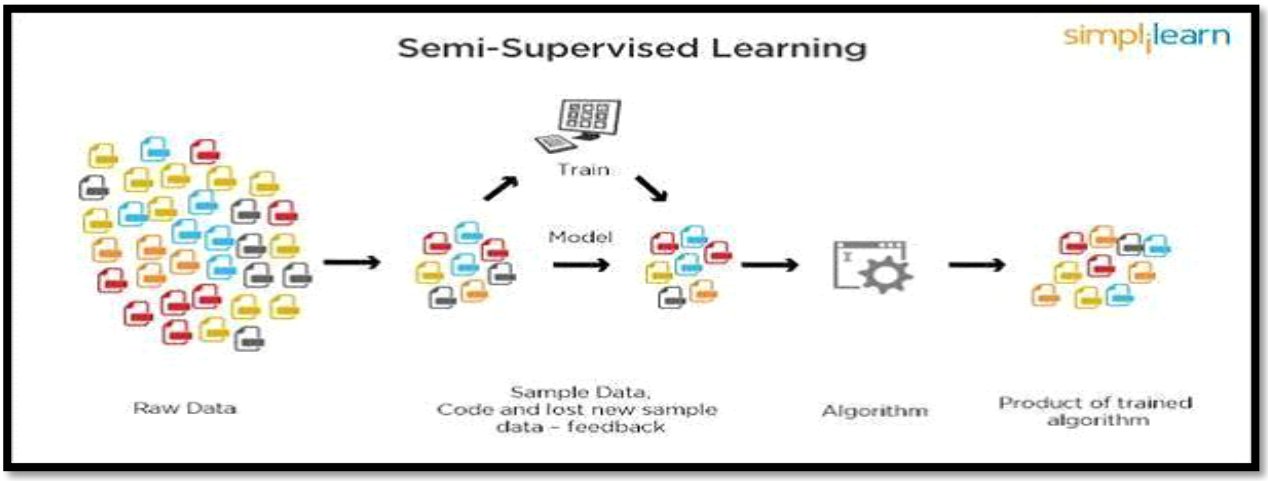

Fig 1.8.3 semi-supervised machine learning

Applications:

- Speech analysis
- Web content classification
- Protein sequence classification

Linear regression:

In Regression, we plot a graph between the variables which best fit the given data points. The machine learning model can deliver predictions regarding the data. In naïve words, *"Regression shows a line or curve that passes through all the data points on a target-predictor graph in such a way that the vertical distance between the data points and the regression line is minimum."* It is used principally for prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables.

Linear regression is a quiet and simple statistical regression method used for predictive analysis and shows the relationship between the continuous variables. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), consequently called linear regression. If there is a single input variable (x), such linear regression is called **simple linear regression**. And if there is more than one input variable, such linear regression is called **multiple**

linear regression. The linear regression model gives a sloped straight line describing the relationship within the variables.

When the value of x (independent variable) increases, the value of y (dependent variable) is likewise increasing. The red line is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best.

1.9 Organization of Document:

Chapter1: Describes motivation, purpose, objective and problem definition are discussed

Chapter2: Explains literature survey and basic concepts and terms.

Chapter3: Discusses about existing and proposed systems.

Chapter4: Explains the analysis of project and system requirements specifications.

Chapter 5: Explains the System design,

Chapter6: Implementation and module explanation of project.

Chapter7: Explains the code functions.

Chapter8: We discuss about testing and validation.

CHAPTER-2
LITERATURE SURVEY

[1] Shahzada Daud , Muti Ullah , Amjad Rehman , Tanzila Saba , Robertas Damasevicius and Abdul Sattar (2023). Online news classification using machine learning techniques.

The main characteristics of the news text in online publications are (1) the prevalence of conversational style; (2) the brevity of the information provided, in which, of course, an information note is a genre of small volume, but the material requires not only the fact of the news that happened but also a detailed description of what is happening and even the author's assessment; And (3) the absence of complicated lexical and grammatical forms. The text performs the functions of informing the audience and influencing it with the help of expressive means. Online news achieves this thanks to simple sentences that do not require the reader to delve into the essence of what is written for a long time.

[2] Rohit Kumar Kaliyar, Kedar Fitwe, Rajarajeswari P, Anurag Goswami (2021). Classification of Hoax/Non-Hoax News Articles on Social Media using an Effective Deep Neural Network.

The internet has become a medium to share information rapidly across the globe. On the other hand, this immense growth of the internet has also led to the sharing of excessive misleading and fake news. There have been many classifiers in the existing literature to classify fraudulent news and legitimate news. Still, they require a lot of data before effective learning, which is present in the most significant cases and it is very challenging to obtain. As deep understanding is one of the emerging technologies, in this paper, we discuss how fake news on the web can be classified using deep learning techniques with testing on several datasets and implementing using several classification models.

[3] Pradeep Kamboj, Lisu Chongtham (2023). Classification of News Articles using Gradient Boosting based Deep Learning Model.

The Newspaper articles offer us insights into several news in today's world. With the popularization of the internet, people are directly influenced by the content they surf on the internet in their daily life and online news is not among an exception. It necessitates the need to classify such news article for its readily availability. In this paper we propose a novel method named Long short-term memory-Gradient boosting (LSTM-GB) deep learning model for classification of news articles and compare the result with existing traditional classifiers. Furthermore, the proposed model attains 99.8% accuracy which outplays other classifiers.

[4] Jeelani ahmed and muqeem ahmed (2021). Online News Classification Using Machine Learning Techniques.

A massive rise in web-based online content today pushes businesses to implement new approaches and resources that might support better navigation, processing, and handling of high-dimensional data. Over the Internet, 90% of the data is unstructured, and there are several approaches through which this data can translate into useful, structured data—classification is one such approach. Classification of knowledge into a good collection of groups is significant and necessary. As the number of machine readable documents proliferates, automatic text classification is badly needed to classify these documents. Unlabeled documents are categorized into predefined classes of labeled documents using text labeling, a supervised learning technique. This paper reviewed some existing approaches for classifying online news articles and discusses a framework for the automatic classification of online news articles. For achieving high accuracy, different classifiers were tried. Our experimental method achieved 93% accuracy using a Bayesian classifier and present in terms of confusion metrics.

CHAPTER-3

EXISTING & PROPOSED SYSTEM

3.1 Existing System:

The features used in the classification of online news articles using optimized machine learning models typically involve representing the content of the articles in a numerical format. Here are some commonly used features:

Bag-of-Words (BoW):

Represents the frequency of words in the document. Each word is treated as an independent feature, and the document is represented as a vector of word frequencies.

Term Frequency-Inverse Document Frequency (TF-IDF):

Combines the frequency of words in a document with their importance in the entire corpus. It helps to highlight words that are specific to a document but not common across all documents.

Word Embeddings:

Dense vector representations of words that capture semantic relationships. Techniques like Word2Vec, GloVe, and FastText are used to convert words into continuous vector spaces.

N-grams:

Represents sequences of adjacent words (bi-grams, tri-grams, etc.) to capture contextual information beyond individual words.

Metadata Features:

Incorporates additional information such as publication date, authorship, source, or any metadata associated with the news articles.

3.2 Proposed System:

Improvising features in the classification of online news

articles can enhance the model's ability to discern subtle nuances in the content and improve overall performance. Here are some feature improvisations:

Embeddings Fusion:

Combine different word embeddings (e.g., Word2Vec, GloVe, BERT) to capture various levels of semantic information within the text.

Attention Mechanisms:

Integrate attention mechanisms to allow the model to focus on important words or phrases in the article, emphasizing key information for classification.

Contextual Embeddings:

Utilize contextual embeddings that consider the surrounding context of each word, offering a more nuanced representation of the semantics within the document.

Domain-Specific Features:

Incorporate domain-specific features related to news articles, such as financial indicators for business news or event timestamps for breaking news.

Multimodal Features:

Integrate features from multiple modalities, such as images, videos, or metadata associated with the news articles, to provide a richer representation of the content.

CHAPTER-4

ANALYSIS

4.1 Purpose of project:

To Classify the online news articles using Machine Learning.

4.2 System Requirements Specifications:

4.2.1 Hardware requirement

- Ram : Min 4GB.
- Processor : Any Update processor
- Hard Disk : Min 40 GB ROM.
- Input Devices : Android's Keyboard
- Output Devices : Mobile

4.2.2 Software requirement

- Operating System : Windows family and android/iOS
- Technology : Python 3.7
- IDE : Jupyter notebook

4.3 Content Diagram of Project:

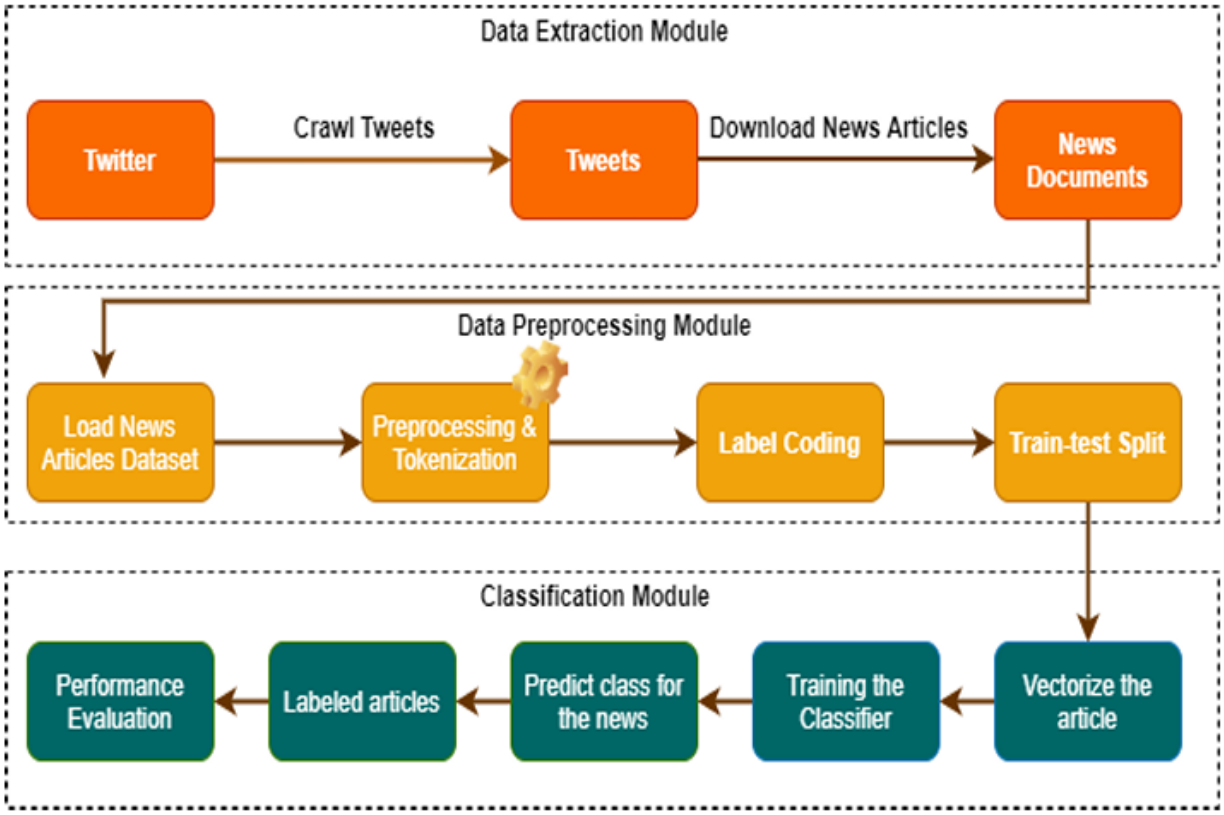

Fig. 4.1 : News article classification process.

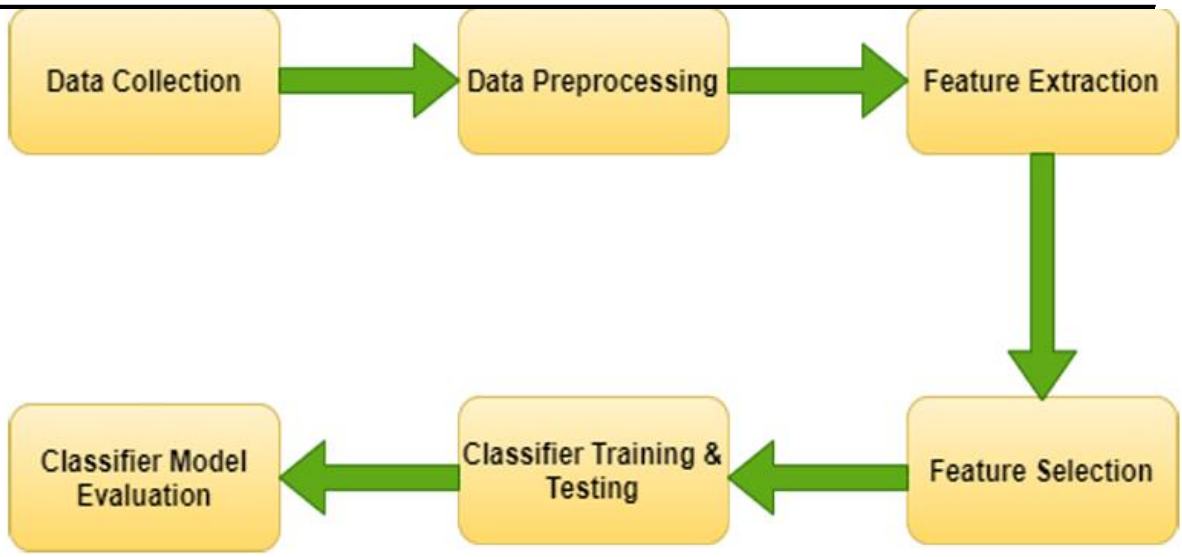

Fig. 4.2 : Text classification process.

4.3.1 Tools and Techniques:

Windows:

Windows is the most widely used operating system for desktop and laptop computers.

Python Idle:

IDLE(short for Integrated Development Environment or Integrated Development and Learning Environment) is an integrated development environment for Python, which has been bundled with the default implementation of the language since 1.5.2b 1. It is packaged as an optional part of the Python packaging with many Linux distributions. It is completely written in Python and the Tkinter GUI tool kit(wrapper functions for Tcl/ Tk).

Based on the powerful editing component Scintilla (free

source code editing component for Win32, GTK+, and Mac OS), Notepad++ is written in C++ and uses pure Win 32 API and STL, which ensures a higher execution speed and smaller program size.

Its main features are:

- Multi-window text editor with syntax highlighting, auto-completion, smart indent, and other.
- Python shell with syntax highlighting.
- Integrated debugger with stepping, persistent breakpoints, and call stack visibility.

IDLE is the integrated development environment (IDE) provided with Python. An IDE combines a program editor and a language environment as a convenience to the programmer. Using IDLE is not a requirement for using Python. There are many other IDLEs that can be used to write Python programs, not to mention a variety of text-based programmer 's editors that many programmers prefer to IDEs.

Overview of Important Features and Tools Provided by PyCharm:

- Code Editor
- Code Navigation
- Refactoring
- Support for Popular Web Technologies

CHAPTER-5
SYSTEM DESIGN

5.1 Introduction:

The goal of this project is to develop a system that can accurately classify online news articles into predefined categories using machine learning models. This system will involve several stages including data collection, preprocessing, feature extraction, model training, evaluation, and deployment.

5.2 Data Collection:

- **Sources**: Online news websites, RSS feeds, APIs (like NewsAPI)
- **Tools**: Web scrapers (e.g., BeautifulSoup, Scrapy), APIs
- **Storage**: Collected data will be stored in a structured format in a database (e.g., MongoDB, PostgreSQL)

5.3 Data Preprocessing:

- **Cleaning**: Removing HTML tags, special characters, and irrelevant information
- **Normalization**: Converting text to lowercase, removing stop words, stemming, and lemmatization
- **Splitting**: Dividing data into training, validation, and test sets

5.4 Feature Extraction

5.4.1 Text Representation:

- TF-IDF (Term Frequency-Inverse Document Frequency)
- Word Embeddings (Word2Vec, GloVe)
- Bag of Words (BoW)
- N-grams

5.4.2 Feature Selection:

- Chi-Square Test
- Information Gain
- Mutual Information

5.5 Model Selection and Training

5.5.1 Machine Learning Models :

- Support Vector Machines (SVM)
- Random Forest (RF)
- Logistic Regression (LR)
- K-Nearest Neighbors (KNN)
- Naïve Bayes (NB)
- Stochastic Gradient Descent (SGD)

5.5.2 Hyperparameter Tuning:

- Grid Search
- Random Search
- Cross-Validation

5.5.3 Training:

- Train models on the training dataset
- Use validation set to fine-tune models and select the best performing model

5.6 Model Evaluation:

5.6.1 Metrics:

- Accuracy
- Precision
- Recall
- F1 Score
- Confusion Matrix

5.6.2 Evaluation Process:

- Evaluate models on the test dataset
- Compare performance metrics across different models
- Select the model with the best overall performance

CHAPTER-6

IMPLEMENTATION

6.1 MODEL BUILDING

6.1.2 PREPROCESSING OF THE DATA :

Preprocessing of the data involves the following steps :

6.1.3 GETTING THE DATASET :

We can get the data set from the database, or we can get the data from client.

6.1.4 IMPORTING THE LIBRARIES :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import pyplot as plt
import seaborn as sn
import spacy
from sklearn.metrics import confusion_matrix
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report
import os
```

Fig 6.1 Importing Libraries

6.1.5 IMPORTING THE DATASET :

Pandas in python provide an interesting method read_csv(). The read_csv function reads the entire dataset from a comma separated values file and we can assign it to a Data Frame to which all the operations can be performed. It helps us to access each row as well as columns and each value can be access using the data frame. Any missing value or NaN value must be cleaned.

6.1.6 READING THE DATASET :

The dataset needs to be imported and read - we use pandas to achieve this,

```
df = pd.read_json('/content/Dataset.zip', lines=True)[['headline','category']]
df.head()
```

Fig 6.2 Reading the Dataset

6.2 VISUALIZATION OF DATASET :

```
df.category.value_counts()
```

```
category
POLITICS          35602
WELLNESS          17945
ENTERTAINMENT     17362
TRAVEL             9900
STYLE & BEAUTY     9814
PARENTING          8791
HEALTHY LIVING     6694
QUEER VOICES       6347
FOOD & DRINK       6340
BUSINESS           5992
COMEDY             5400
SPORTS             5077
BLACK VOICES       4583
HOME & LIVING      4320
PARENTS            3955
THE WORLDPOST      3664
WEDDINGS           3653
WOMEN              3572
CRIME              3562
IMPACT             3484
DIVORCE            3426
WORLD NEWS         3299
MEDIA              2944
WEIRD NEWS         2777
GREEN              2622
WORLDPOST          2579
RELIGION           2577
STYLE              2254
SCIENCE            2206
TECH               2104
TASTE              2096
MONEY              1756
ARTS               1509
ENVIRONMENT        1444
FIFTY              1401
GOOD NEWS          1398
U.S. NEWS          1377
ARTS & CULTURE     1339
COLLEGE            1144
LATINO VOICES      1130
CULTURE & ARTS     1074
EDUCATION          1014
Name: count, dtype: int64
```

Fig 6.2.1 Data category counts

CODE :

Importing Libraries :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import pyplot as plt
import seaborn as sn
import spacy
from sklearn.metrics import confusion_matrix
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report
import os
```

Reading the jason file :

```
df = pd.read_json('/content/Dataset.zip', lines=True)[['headline','category']]
df.head()
```

Checking the category counts :

```
df.category.value_counts()
```

Select only rows where the category is in a list of desired values :

```
desired_categories = ['CRIME', 'COMEDY', 'WEDDINGS', 'SPORTS']
df_new = df[df['category'].isin(desired_categories)]
df_new.head()
```

New category counts :

```
df_new.category.value_counts()
```

Balncing with educating articles:

```
min_samples = 3562 # we have these many EDUCATION articles
df_business = df_new[df_new.category=="COMEDY"].sample(min_samples, random_state=2022)
df_sports = df_new[df_new.category=="SPORTS"].sample(min_samples, random_state=2022)
df_crime = df_new[df_new.category=="CRIME"].sample(min_samples, random_state=2022)
df_weddings = df_new[df_new.category=="WEDDINGS"].sample(min_samples, random_state=2022)
```

```
df_balanced = pd.concat([df_business,df_sports,df_crime,df_weddings],axis=0)
df_balanced.category.value_counts()
```

target = {'COMEDY': 0, 'SPORTS': 1, 'CRIME': 2, 'WEDDINGS': 3}

```
df_balanced['category_num'] = df_balanced['category'].map({'COMEDY': 0,'SPORTS': 1, 'CRIME': 2, 'WEDDINGS': 3})
```

```
df_balanced.tail()
```

Training dataset :

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df_balanced.headline, df_balanced.category_num, test_size=0.2,)
```

```
print(X_train.shape)
X_train.head()
```

```
y_train.value_counts()
```

```
y_test.value_counts()
```

Vectorization (Using 1-gram) :

```
clf = Pipeline([('vectorizer', CountVectorizer()),('Multi NB', MultinomialNB())])
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
```

Using 1-gram and bigrams :

```
#1. create a pipeline object
clf = Pipeline([('vectorizer_1_2_gram', CountVectorizer(ngram_range = (1, 2))),('Multi NB', MultinomialNB())])
clf.fit(X_train, y_train)
#3. get the predictions for X_test and store it in y_pred
y_pred = clf.predict(X_test)
#4. print the classfication report
print(classification_report(y_test, y_pred))
```

Using 1-gram to trigrams :

```
#1. create a pipeline object
clf = Pipeline([('vectorizer_1_3_grams', CountVectorizer(ngram_range = (1, 3))),('Multi NB', MultinomialNB())])
#2. fit with X_train and y_train
clf.fit(X_train, y_train)
#3. get the predictions for X_test and store it in y_pred
y_pred = clf.predict(X_test)
#4. print the classfication report
print(classification_report(y_test, y_pred))
```

Use text pre-processing to remove stop words, punctuations and apply lemmatization :

```
nlp = spacy.load("en_core_web_sm")
def preprocess(text):
    doc = nlp(text)
    filtered_tokens = []
    for token in doc:
        if not token.is_stop and not token.is_punct:
            filtered_tokens.append(token.lemma_)
    return " ".join(filtered_tokens)
```

```
df_balanced['preprocessed_txt'] = df_balanced['headline'].apply(preprocess)
df_balanced.head()
```

```
X_train, X_test, y_train, y_test = train_test_split(df_balanced.preprocessed_txt, df_balanced.category_num,test_size=0.2,
random_state=2023,stratify=df_balanced.category_num)
```

```
print(X_train.shape)
X_train.head()
```

```
#1. create a pipeline object
clf = Pipeline([('vectorizer_bow', CountVectorizer(ngram_range = (1, 2))),('Multi NB', MultinomialNB())])
#2. fit with X_train and y_train
clf.fit(X_train, y_train)
#3. get the predictions for X_test and store it in y_pred
y_pred = clf.predict(X_test)
#4. print the classfication report
print(classification_report(y_test, y_pred))
```

Printing Confusion matrix :

```
cm=confusion_matrix(y_test,y_pred)
cm
```

Figure of Confusion matrix :

```
plt.figure(figsize = (10,7))
sn.heatmap(cm, annot=True)
plt.xlabel('Prediction')
plt.ylabel('Truth')
```

Bar Graph for actual and predicted values :

```
# Define the confusion matrix
matrix = np.array([
    [580, 42, 34, 58],
    [47, 620, 32, 13],
    [16, 19, 660, 12],
    [18, 9, 9, 680]
])

# Calculate the total actual values for each class
actual_values = matrix.sum(axis=1)

# Calculate the total predicted values for each class
predicted_values = matrix.sum(axis=0)

# Define the class labels
class_labels = ['Class 0', 'Class 1', 'Class 2', 'Class 3']

# Plot the actual vs. predicted values
x = np.arange(len(class_labels))  # the label locations

fig, ax = plt.subplots()
width = 0.35  # the width of the bars

rects1 = ax.bar(x - width/2, actual_values, width, label='Actual')
rects2 = ax.bar(x + width/2, predicted_values, width, label='Predicted')

# Add some text for labels, title, and custom x-axis tick labels, etc.
ax.set_xlabel('Class')
ax.set_ylabel('Count')
ax.set_title('Actual vs Predicted Counts by Class')
ax.set_xticks(x)
ax.set_xticklabels(class_labels)
ax.legend()
```

```
# Attach a text label above each bar in rects, displaying its height.
def autolabel(rects):
    """Attach a text label above each bar in rects, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3),  # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)

fig.tight_layout()

plt.show()
```

Pie Chart :

```
# Define the confusion matrix
matrix = np.array([
    [580, 42, 34, 58],
    [47, 620, 32, 13],
    [16, 19, 660, 12],
    [18, 9, 9, 680]
])

# Calculate the total actual values for each class
actual_values = matrix.sum(axis=1)

# Calculate the total predicted values for each class
predicted_values = matrix.sum(axis=0)

# Define the class labels
class_labels = ['Class 0', 'Class 1', 'Class 2', 'Class 3']

# Plot pie charts for actual and predicted values
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 7))

# Actual values pie chart
ax1.pie(actual_values, labels=class_labels, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)
ax1.set_title('Actual Values Distribution')

# Predicted values pie chart
ax2.pie(predicted_values, labels=class_labels, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)
ax2.set_title('Predicted Values Distribution')

plt.show()
```

CHAPTER-7
TEST CASES

7 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

Testing Objectives:

- To ensure that during operation the system will perform as per specification.
- To make sure that system meets the user requirements during operation.
- To make sure that during the operation, incorrect input, processing and output will be detected.
- To see that when correct inputs are fed to the system the outputs are correct.
- To verify that the controls in corporate in the same system as intended.
- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding a yet undiscovered error.

7.1 Types Of Testing:

We have different types of testing methodologies. They are:

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Acceptance Testing

7.1.1 Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and program inputs produce valid outputs.

7.1.2 Integration Testing:

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2. Functional Testing:

Functional tests provide systematic demonstrations that functions tests are available as specified by the business and technical requirements, system documentation, and user manuals.

7.3 System Testing:

System testing ensures that the entire integrated software system meets requirements.

- **White Box Testing:** It is a testingin which the software tester has knowledge of the inner workings, structure, and language of the software.
- **Black Box Testing:** It is testing the software without any knowledge of the inner workings, structure, or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source documents, such as a specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. We cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.4 Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

7.5 RESULT ANALYSIS :

7.5.1 Model performance :

	precision	recall	f1-score	support
0	0.88	0.81	0.84	712
1	0.90	0.87	0.88	713
2	0.90	0.93	0.92	712
3	0.89	0.95	0.92	713
accuracy			0.89	2850
macro avg	0.89	0.89	0.89	2850
weighted avg	0.89	0.89	0.89	2850

Fig 7.5.1

7.5.2 Confusion Matrix :

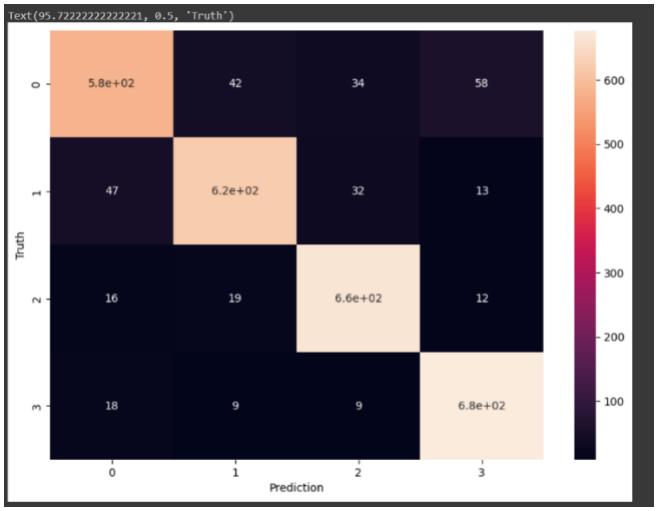

Fig 7.5.2

7.5.3 Accuracy of Models :

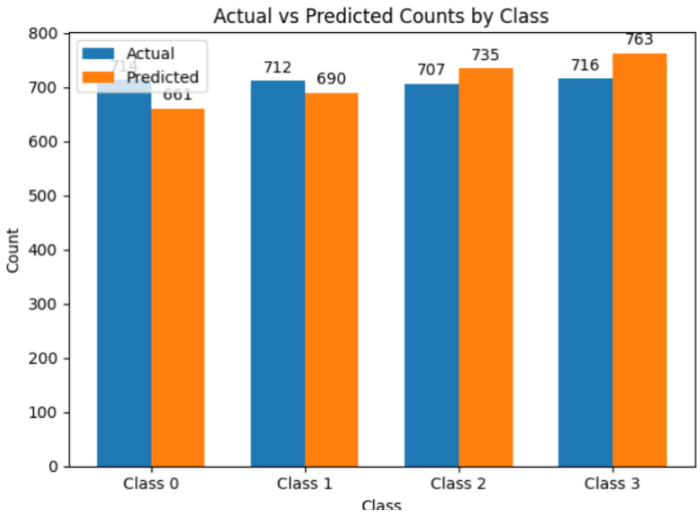

Fig 7.5.3

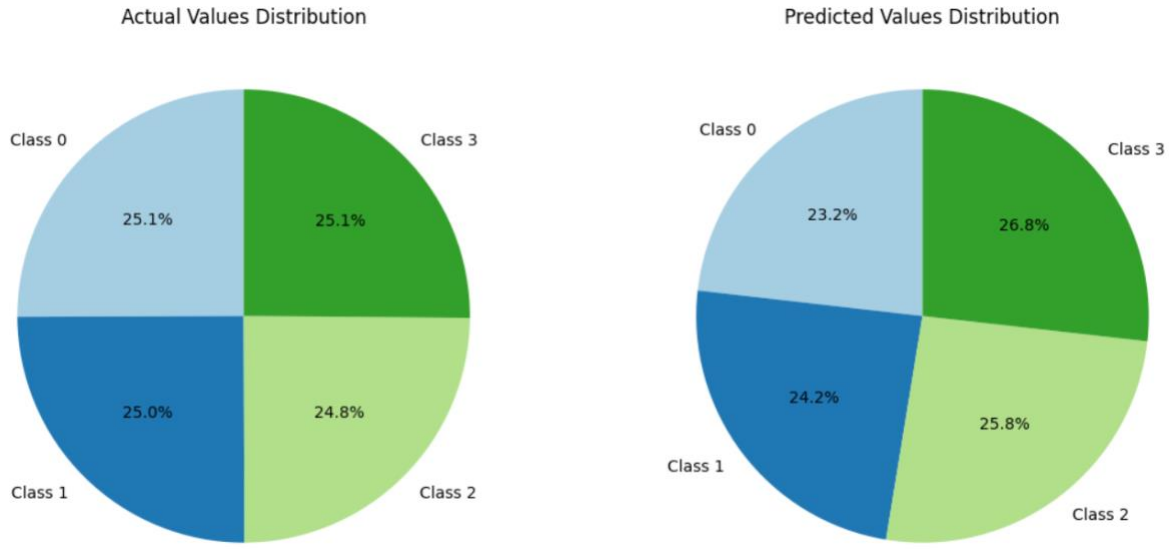

Fig 7.5.4

CONCLUSION :

We conclude that , The Classification of Online News Articles using Optimized Machine Learning represents a significant advancement in information organization and retrieval in the digital age. Through the development and implementation of a robust machine learning system, this project has successfully demonstrated the effectiveness of utilizing optimization techniques to enhance classification accuracy and performance.

This project highlights the potential of machine learning in enhancing the organization and retrieval of digital information. It sets the foundation for future advancements, such as integrating more sophisticated natural language processing techniques, expanding the range of topics, and continuously improving model accuracy. The successful implementation of this project not only contributes to the field of machine learning but also offers valuable insights into the development of intelligent content management systems.

FUTURE ENHANCEMENT

All the considered machine learning methods' accuracies are compared based on which one prediction model is generated. Hence, the aim is to use various evaluation metrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts efficiently.

In today's fast-paced digital world, the abundance of online news articles poses a challenge in efficiently accessing relevant information . Classifying online news using machine learning provides several benefits, such as enabling efficient content categorization, automating information retrieval, and enhancing personalized recommendations. We add more features to the present model to increace the efficiency and performance.

REFERENCES/BIBLIOGRAPHY

[1] Mitchell, A.; Rosenstiel, T. Navigating News Online: Where People Go, How They Get There and What Lures Them Away. PEWResearch Center's Project for Excellence in Journalism. 2011.

[2] Harouni, M.; Rahim, M.S.M.; Al-Rodhaan, M.; Saba, T.; Rehman, A.; Al-Dhelaan, A. Online Persian/Arabic script classification without contextual information. Imaging Sci. J. 2014, 62, 437–448.

[3] Bakshy, E.; Rosenn, I.; Marlow, C.; Adamic, L. The Role of Social Networks in Information Diffusion. In Proceedings of the WWW 2012: 21st World Wide Web Conference, Lyon, France, 16–20 April 2012; pp. 519–528.

[4] Bennett, W.L.; Iyengar, S. A New Era of Minimal Effects? The Changing Foundations of Political Communication. J. Commun. 2008, 58, 707–731.

[5] Rehman, A.; Saba, T. Off-line cursive script recognition: Current advances, comparisons and remaining problems. Artif. Intell. Rev. 2012, 37, 261–288.

[6] Kull, S.; Ramsay, C.; Lewis, E. Media, Misperceptions, and the Iraq War. Polit. Sci. Q. 2003, 118, 569–598

[7] Chen, Z.Q.; Zhang, G.X. Survey of text mining, Pattern Recognit. Artif. Intell. 2005, 18, 65–74.

[8] Schutze, H.; Manning, C.D.; Raghavan, P. Introduction to Information Retrieval; Cambridge University Press: Cambridge, UK, 2008.

[9] Javed, R.; Rahim, M.S.M.; Saba, T.; Rehman, A. A comparative study of

features selection for skin lesion detection from dermoscopic images. Netw. Model. Anal. Health Inform. Bioinform. 2020, 9, 1–13. [CrossRef]

[10] Larabi-Marie-Sainte, S.; Aburahmah, L.; Almohaini, R.; Saba, T. Current Techniques for Diabetes Prediction: Review and Case Study. Appl. Sci. 2019, 9, 4604. [CrossRef]

[11] Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 2011, 12, 2493–2537.

[12] Computers 2023, 12, 16 15 of 16 13. Rehman, A.; Saba, T. Performance analysis of character segmentation approach for cursive script recognition on benchmark database. Digit. Signal Process. 2011, 21, 486–490.

[13] Tesfagergish, S.G.; Kapo˙ ci¯ ut˙ e-Dzikien˙ e, J.; Damaševi˘ cius, R. Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning. Appl. Sci. 2022, 12, 8662.

[14] Saba, T.; Rehman, A.; Altameem, A.; Uddin, M. Annotated comparisons of proposed preprocessing techniques for script recognition. Neural Comput. Appl. 2014, 25, 1337–1347.

[15] Dalyan, T.; Ayral, H.; Özdemir, Ö. A Comprehensive Study of Learning Approaches for Author Gender Identification. Inf. Technol. Control 2022, 51, 429–445.

[16] Shambour, Q.Y.; Abu-Shareha, A.A.; Abualhaj, M.M. A Hotel Recommender System Based on Multi-Criteria Collaborative Filtering. Inf. Technol. Control 2020, 51, 390–402.

[17] Wei, W.; Wang, Z.; Fu, C.; Damaševiˇ cius, R.; Scherer, R.; Wożniak, M. Intelligent recommendation of related items based on naive bayes and collaborative filtering combination model. J. Phys. Conf. Ser. 2020, 1682, 012043.

[18] Tesfagergish, S.G.; Damaševiˇcius, R.; Kapoˇci¯utˇe-Dzikienˇe, J. Deep fake recognition in tweets using text augmentation, word embeddings and deep learning. In Computational Science and Its Applications, ICCSA 2021; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12954, pp. 523–538.

[19] Jiang, M.; Zou, Y.; Xu, J.; Zhang, M. GATSum: Graph-Based Topic-Aware Abstract Text Summarization. Inf. Technol. Control 2022, 51, 345–355.

[20] Jiang, M.; Zou, Y.; Xu, J.; Zhang, M. GATSum: Graph-Based Topic-Aware Abstract Text Summarization. Inf. Technol. Control 2022, 51, 345–355.

[21] Kaur, Gurmeet, and Karan Bajaj. "News classification and its techniques: a review." *IOSR Journal of Computer Engineering* 18.1 (2016): 22-26.

[22] Kaur, G., & Bajaj, K. (2016). News classification and its techniques: a review. *IOSR Journal of Computer Engineering*, *18*(1), 22-26.

[23] Rana, Mazhar Iqbal, Shehzad Khalid, and Muhammad Usman Akbar. "News classification based on their headlines: A review." *17th IEEE International Multi Topic Conference 2014*. IEEE, 2014.

[24] Ahmed, Jeelani, and Muqeem Ahmed. "Online news classification using machine learning techniques." *IIUM Engineering Journal* 22.2 (2021): 210-225.

[25] Barberá, Pablo, et al. "Automated text classification of news articles: A practical guide." *Political Analysis* 29.1 (2021): 19-42.

[26] Doddi, Kiran S., Y. V. Haribhakta, and Parag Kulkarni. "Sentiment classification of news article." *Diss. College of Engineering Pune* 70 (2014).

[27] Jang, Beakcheol, Inhwan Kim, and Jong Wook Kim. "Word2vec convolutional neural networks for classification of news articles and tweets." *PloS one* 14.8 (2019): e0220976.

[28] Cooley, Robert. "Classification of news stories using support vector machines." *Proc. 16th International Joint Conference on Artificial Intelligence Text Mining Workshop*. Vol. 26. 1999.

[29] CHASE, Zach, Nicolas Genain, and Orren Karniol-Tambour. "Learning Multi-Label Topic Classification of News Articles." (2014).

[30] Sebők, Miklós, and Zoltán Kacsuk. "The multiclass classification of newspaper articles with machine learning: The hybrid binary snowball approach." *Political Analysis* 29.2 (2021): 236-249.

[31] Dilrukshi, Inoshika, Kasun De Zoysa, and Amitha Caldera. "Twitter news classification using SVM." *2013 8th International Conference on Computer Science & Education*. IEEE, 2013.

[32] Asad, Muhammad Imran, et al. "Classification of News Articles using Supervised Machine Learning Approach." *Pakistan Journal of Engineering and Technology* 3.03 (2020): 26-30.

[33] Kayakuş, Mehmet, and Fatma Yiğit Açıkgöz. "Classification of news texts by categories using machine learning methods." *Alphanumeric Journal* 10.2 (2022): 155-166.

[34] Carreira, Ricardo, et al. "Evaluating adaptive user profiles for news classification." *Proceedings of the 9th international conference on Intelligent user interfaces*. 2004.

[35] Zhang, Menghan. "Applications of deep learning in news text classification." *Scientific Programming* 2021.1 (2021): 6095354.

[36] Joo, Kil-Hong, et al. "Hierarchical automatic classification of news articles based on association rules." *Journal of Korea Multimedia Society* 14.6 (2011): 730-741.

[37] Ahmed, Jeelani, and Muqeem Ahmed. "Online news classification using machine learning techniques." IIUM Engineering Journal 22.2 (2021): 210-225.

[38] Brand, Dirk, and Brink Van Der Merwe. "Comment classification for an online news domain." (2014).

[39] Sebők, Miklós, and Zoltán Kacsuk. "The multiclass classification of newspaper articles with machine learning: The hybrid binary snowball approach." Political Analysis 29.2 (2021): 236-249.

[40] Qian, Yu, et al. "On detecting business event from the headlines and leads of massive online news articles." Information Processing & Management 56.6 (2019): 102086.

[41] Fagbola, Temitayo Matthew, Colin Surendra Thakur, and Oludayo

Olugbara. "News article classification using Kolmogorov complexity distance measure and artificial neural network." International Journal of Technology 10.4 (2019): 710-720.

[42] Pierce, Matthew. "Large-scale multi-label text classification for an online news monitoring system." Computer Science 77 (2015): 0.