MAJOR PROJECT REPORT – PHASE I
On

Newspaper Article Classification Using NLP

submitted to

BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE AND ENGINEERING
by

KOLTHURI SPANDAN REDDY	19241A0580
GANJI VISHNUVARDHAN REDDY	19241A0571
ANIKETH DESHMUKH	19241A0569
NALLA KSHITHIJ REDDY	19241A0688

under the esteemed guidance of

Dr. S. GOVIND RAO
Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Gokaraju Rangaraju Institute of Engineering and Technology

(Autonomous)

Bachupally, Hyderabad, Telangana

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Autonomous)
Bachupally, Hyderabad, Telangana - 500090

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the major project titled "**Newspaper Article Classification using NLP**" is a bonafide work done by **Kolthuri Spandan Reddy** (19241A0580), **Ganji Vishnuvardhan Reddy** (19241A0571), **Aniketh Deshmukh** (19241A0569), **Nalla Kshithij Reddy** (9241A0588) in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, Hyderabad and that this work has not been submitted for the award of any other Degree/Diploma of any Institution/University.

Section Project Coordinator Project Guide

DECLARATION

We hereby declare that the project titled **"Newspaper Article Classification using NLP"** is original and bonafide work of our own in the partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, submitted to the Department of Computer Science and Engineering, GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY(Autonomous) Hyderabad under guidance of **Dr. S. Govind Rao,** Professor and has not been copied from any earlier reports/works.

Kolthuri Spandan Reddy	19241A0580
Ganji Vishnuvardhan Reddy	19241A0571
Aniketh Deshmukh	19241A0569
Nalla Kshithij Reddy	19241A0588

 GANJI VISHNUVARDHAN REDDY Roll No: 19241A0571	 KOLTHURI SPANDAN REDDY Roll No: 19241A0580	 ANIKETH DESHMUKH Roll No: 19241A0569	 NALLA KSHITHIJ REDDY Roll No: 19241A0588

ACKNOWLEDGEMENT

We are very grateful to our Project guide **Dr. S. Govind Rao**, Professor, Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous), for his extensive patience and guidance throughout our project work.

We are very grateful to our Project Co-ordinator **Mrs. V. Jyothi,** Assistant. Professor, Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous), for extending his support and assisting us throughout our project work.

We would like to thank **Dr. K. Madhavi**, Head of the Department, Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous), for having provided the freedom to use all the facilities available in the department, especially the laboratories and library.

We also express a deep sense of gratitude to our **Director Sir** & **Principal Sir**, for having provided us with adequate facilities to pursue our project and for providing us the conducive environment for carrying through our academic schedules and projects with ease.

We sincerely thank all the staff of the Department of Computer Science and Engineering, for their timely suggestions, healthy criticism and motivation during the course of our project work.

Finally, we express our immense gratitude with pleasure to all who have either directly or indirectly contributed to our need at right time for the development and success of our project work

Kolthuri Spandan Reddy	**19241A0580**
Ganji Vishnuvardhan Reddy	**19241A0571**
Aniketh Deshmukh	**19241A0569**
Nalla Kshithij Reddy	**19241A0588**

ABSTRACT

In many real-world scenarios, the ability to automatically classify news-paper articles into a fixed set of categories (sports, politics, tech etc.) is highly beneficial. News article applications can directly classify the articles into a set of categories and recommend to users based on their interests. Natural Language Processing offers powerful techniques which can be used for classification of various documents. These techniques are predicted on the hypothesis that documents in different categories distinguish themselves by features of the language contained in each document. Salient features for document classification may include word structure, word frequency, and natural language structure in each document.

We already have a large archive of already classified articles, so we are able to make use of supervised classification techniques. Classification algorithms such as Multinomial Naive Bayes, Logistic Regression, Random Forest Classifier etc, can be used to classify the articles. Famous python packages like Natural Language Toolkit (NLTK), Scikit-Learn can also be used.

Table of Contents

INTRODUCTION

The first newspaper was published in the United States in 1704. It was called Public Occurrences, Both Foreign and Domestic. The first newspaper in the world was published on March 25, 1665. It was called The London Gazette and it contained news of events happening in England and abroad. After that the newspaper was circulated to all over the world. Present newspaper consists of different sections and providing much information that is happening all over the world. The current newspaper consists of different sections which focuses on different issues that are happening all round the globe.

News Paper is a paper that contains information about many things that are happening around the world. It consists of different sections like Politics, Real estate, Entertainment, Studies, Business, Sports, and current issues. Politics section contains information regarding the present, past and future information of politics. The newspaper provide the schemes that are providing by the government to the people and updates of all schemes that are ongoing presently. It also consists of the following information about major issues of the society that is currently concern about people. Classifieds section consists of different advertisement about the jobs, marriage proposals, rentals and properties that are available for the lease or sale. Entertainment section consists of information about movies, concerts, and other information about the entertainments. Sports section consists of information about the sports events, locations and results of sports matches etc.. Business section consists of information about the share market, trading, new companies, and their products.

Earlier newspaper is circulated all over the places in the form of physical paper. The paper was printed in the warehouses and distributed all over the world for circulation. Printing the newspaper and circulation around the world early in the morning is a big task for the newspaper distributers. After the digitalization and evaluation of smart phones and internet in the world, now the newspaper uploading to the web so that every subscriber is easily accessing the newspaper from their phone or computer using the internet. This made people for easy access to paper and distributors do not have any problem for delivery of the paper.

Newspaper has different sections that provide information about different topic but

there is a problem with the newspaper. Newspaper consists of the more sections, so user must search for his interesting section in the newspaper. This process is a very time consuming, so it is wasting the user time. To solve this problem, we have to divide the newspaper into different sections to directly access the one section in the newspaper. There are different algorithms are there to solve this problem. We use Natural Language Processing to distinguish the news articles into different classifications such as Politics, General Knowledge, Sports, Entertainment, Business etc. so the newspaper applications will only show the user interest articles of the daily newspaper.

TECHNOLOGY

Our project is completely built in the Python using Visual Studio Code IDE. Python has many inbuilt libraries that are easy to use and reduce the time of implementation. Python has many inbuilt libraries, but we are using only some of them to develop our model. “The python libraries used in our project are numpy, pandas, sklearn, matplotlib, mpl_toolkits etc.” Each library has its own specific set of usage in the development of the project.

1) Python:

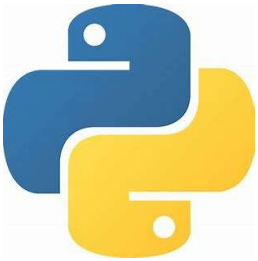
“Python is a high – level programming language used to develop software’s and web pages”. It is an interpreted language. Interpreted language means “the source code is not directly translated by the target machine instead a different program like interpreter reads and executes the code.” Python usually has less code than the many other languages which makes programmer to develop the software easily. Most of the functional works are generally done by using the inbuilt functions of python. Python has large set of inbuilt libraries; they are most useful for developing new models. Some of the libraries used in machine learning projects are OpenCV, Keras, Pandas, NumPy, matplotlib, Sklearn, TensorFlow etc. Each library has different functions that reduce the work for the programmer.

Installing python:

sudo apt get install python

checking python version:

python3 -V



2) NumPy:

Python do not have the array data structure in it, so we have import NumPy from the libraries to use the arrays data structure. NumPy has some other data structure like matrices. NumPy is useful for working with the linear algebra. NumPy is used to perform different operations on the arrays or matrices. NumPy also has trigonometric functions like cos, tan, sin etc. In our project NumPy is used to store the stock prices of the different companies and to calculate the average values of stock data.

For installing numpy we must install pip first:

sudo apt install python3-pip

Installing NumPy in python:

pip install numpy

Importing Numpy:

Import numpy as np

3) Pandas:

“Pandas is used for data analysis and associated manipulation of tabular data in Data frames. Pandas allow importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel.”

Pandas is used to handle the missing data and useful for slicing, merging, concatenation of the data.

Installing pandas in python:

pip install pandas

importing pandas:

import pandas as pa

4) NLTK:

Making genuine human language accessible to computer systems is the goal of the area of natural language processing (NLP). You may use the Python library NLTK, or Natural Language Toolkit, for NLP. A generous portion of the data that you could be examining is unstructured and contains text that can be read by humans. The Natural Language Toolkit (NLTK) is a Python programming environment for creating applications for statistical natural language processing (NLP). For tokenization, parsing, classification, stemming, tagging, and semantic reasoning, it includes text processing libraries.

Installing NLTK in python:

pip install nltk

importing NLTK:

```
import nltk as nl
```

5) Spacy:

SpaCy is a Python NLP library that is open-source and free. It is created to produce information extraction or natural language processing systems and is built in Cython. It offers a clear and approachable API and is designed for use in production. SpaCy enables you to create applications that manage and "understand" massive amounts of text because it is made primarily for usage in production environments. It can be used to create systems for information extraction, NLU, or text pre-processing before deep learning.

Installing spacy in python:

```
pip install spaCy
```

importing spacy:

```
import spaCy as sp
```

6)SKLearn:

The most useful machine learning library in Python is scikit-learn or sklearn. Sklearn is the most user-friendly and reliable machine learning software. Dimensionality reduction, classification, clustering, cross validation, regression, feature extraction, and feature selection are just a few of the machine learning and statistical modelling capabilities provided by the sklearn toolbox. The sklearn package includes several supervised and unsupervised algorithms. Instead of importing, modifying, and summarizing data, the sklearn toolbox focuses primarily on data modelling. It provides simple and effective approaches for predictive data analysis. The sklearn library is a simple and effective tool for predicting consumer behavior, creating neuroimages, and more. As a result, it is simple to use.

Installing sklearn in python:

```
pip install sklearn
```

importing sklearn:

```
import sklearn as skl
```

7)Matplotlib:

Matplotlib is a cross-platform data visualization and graphical charting programmed for Python and its numerical extension NumPy. Numpy is a required component of matplotlib, which makes use of numpy algorithms to manage numerical data and multi-dimensional arrays. Matplotlib utilizes pandas for data manipulation and analysis, however it is not a must like numpy. Static, interactive, and animated visualizations are all possible with this Python library. Matplotlib makes difficult things possible and easy things simple. Matplotlib makes it simple to

create basic graphs with one-liners. Matplotlib provides an object-oriented API for embedding charts in Python GUI toolkit-based applications.

Installing Matplotlib in python:

Pip install matplotlib

Importing Matplotlib:

Import matplotlib as mpl

CHAPTER -2
LITERATURE SURVEY

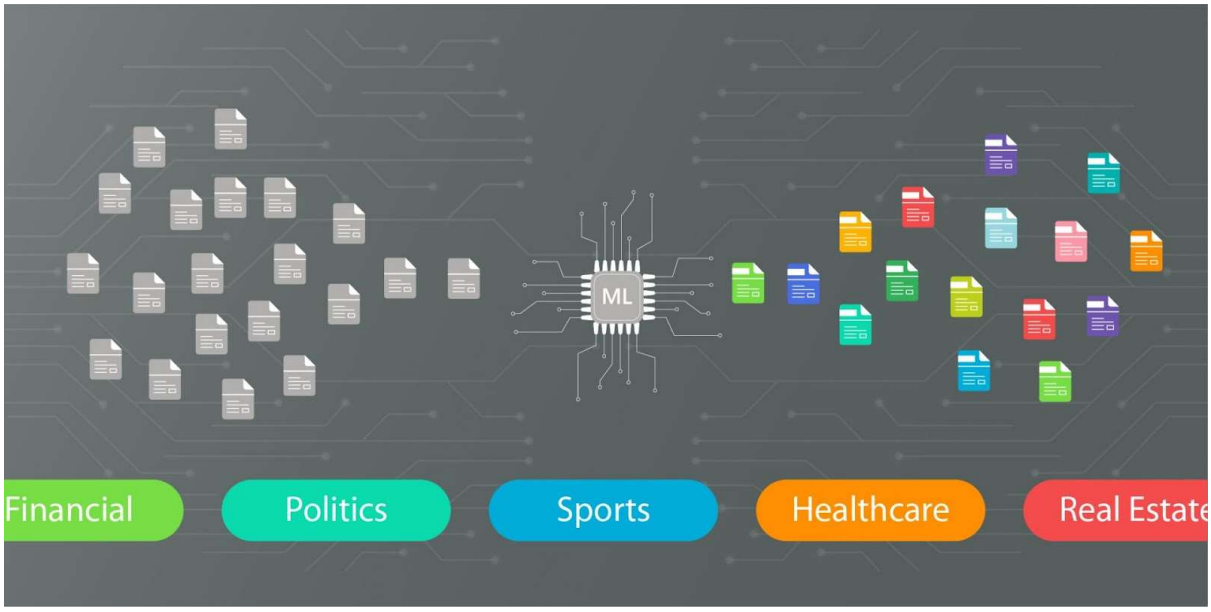
Existing System

Naive Bayes Algorithm

Here we use normal Naive Bayes Classifier to classify the News articles into different categories. We can classify the news articles in different categories like sports, politics, science, entertainment, technology etc. We use Natural Language Processing (NLP) to classify the news articles.

Data Pre-processing

Initially before applying the algorithm, we have to pre-process the data to ease the processing. First, we will convert all the capital letters of the text to lower case letters. Now, we have to remove all the stop words from the text. Stop words are those words in the text, which generally does not have significant meaning. The next step of data pre-processing is Lemmatizing. Lemmatization groups together different forms of a word so that they can be considered as one group. The text is also broken down into tokens.



Bag of Words

The text is represented as a vector of words using Bag of Words. The Bag-of-Words model is a way of representing the text in the form of a vector. It is a simplifying representation used in Natural Language processing and Information retrieval.

In this model, a text is represented as the bag of its words, disregarding grammar and even order but keeping multiplicity. Here the structure and order of the words are ignored.

This model is mainly based on the frequency of the words i.e., the more the frequency of word belonging to a certain category, the more the chances of the article to align to that category.

We use the below formula in the algorithm to classify the news article

$$c_j = \arg\max_{c \in C} P(C) \prod_{i=1}^{n} P(a_i|C)$$

Were,
Cj = maximum probability
C = class of the category
Ai = attribute

Advantages of using Naive Bayes Classifier:

- It has a simple implementation.
- I can be used on both continuous and discrete data.
- It does not require so much training data.
- It is fast and can be used to make real-time predictions.

Disadvantages of using Naive Bayes Classifier:

- In this model we classify based on the frequency of the words, but some words have more weightage in classifying the article.
- It is very simple but in the real world there are other factors which should be considered in classifying the articles.

Result

The model is good enough to classify the few categories of news articles but can be improved to classify into more categories and subcategories articles.

Proposed System

Naive Bayes classifier with Weights attached to the Words

This model is similar to the Simple Naive Bayes Classifier but here we attached a certain weight to the words, so that it would be easier for us to classify the articles into different categories. Here parts of Speech are identified, and semantics of the words considered in sentences.

Initially we start with pre-processing and cleaning the data. It contains the steps like removing stop words, Lemmatizing etc.

Then we apply the below formula in the algorithm

$$c_j = \arg\max_{c \in C} P(C) \prod_{i=1}^{n} P(a_i|C)^{w_i}$$

Were,
Cj = maximum probability
C = class of the category
Ai = attribute
Wi = weight of the attribute

Advantages of using Naive Bayes Classifier with Weights attached to words

- It is simple to use.
- It is more accurate than the Simple Naive Bayes Algorithm.
- It is fast and can be used to make real-time predictions.

Result

Here we have used a Naive Bayes Classifier with weights attached to the words. It solves the problem of different words having different semantical weights.

CHAPTER – 3
SYSTEM SPECIFICATIONS

SCOPE OF THE PROJECT

Newspaper Classification is mostly used in the news applications where the applications will only show the news that is requested by the end user. For example, some people like sports news and some other like the tech news and some other like the politics. So instead of showing all types of news to the people the model will show only preferred or liked articles to the users.

SYSTEM REQUIREMENT SPECIFICATIONS

Software and Hardware Requirements:

1) **Software Requirements:**

We require the following python libraries-

1. NumPy - v1.17.4

2. Pandas – v0.25.3
3. Sklearn – v1.5.2
4. Matplotlib – 1.2.2
5. Mpl_toolkits – v3.5.0

2) Hardware Requirements:

We require the following Hardware:

1. Operating System: Windows 7 or later
2. CPU: Intel core i3 or greater
3. RAM: Minimum 4GB

Feasibility study:

1)Technical feasibility:

The resources for the project are easily available. Also, the project does not require high end hardware and software. The code is memory efficient and fast. The algorithm developed is significant for a long period of time. The code is written in python which is easy to understand and uses advanced machine learning techniques.

2)Economic feasibility:

The algorithm developed during the research are highly economical as it uses limited resources. The data used for training and testing the model are easily available online and the execution of the proposed model can be done anywhere making it economically feasible. The proposed model is tested to get correct in most of the cases proving it to be economically favorable. To run the model, we have feed the data in real time and gives the output. This makes the proposed model to operate with low cost.

3)Operational feasibility:

The proposed model is very feasible to operate as it requires very less human

interference. We just have to feed the model with the data which makes it easily operable. The model has been trained and tested using already existing data from online resources. The model has been successfully tested and need not be modified frequently making it highly reliable. The model will work significantly good for long period as per the current market trends. This makes the proposed model feasible to operate.

4)Time feasibility:

The proposed model detects the news articles very fast to save the time for both the user and the news applications. This model works independently without the human intervention that means independently to provide accurate news articles of the user preferences with in very less time.

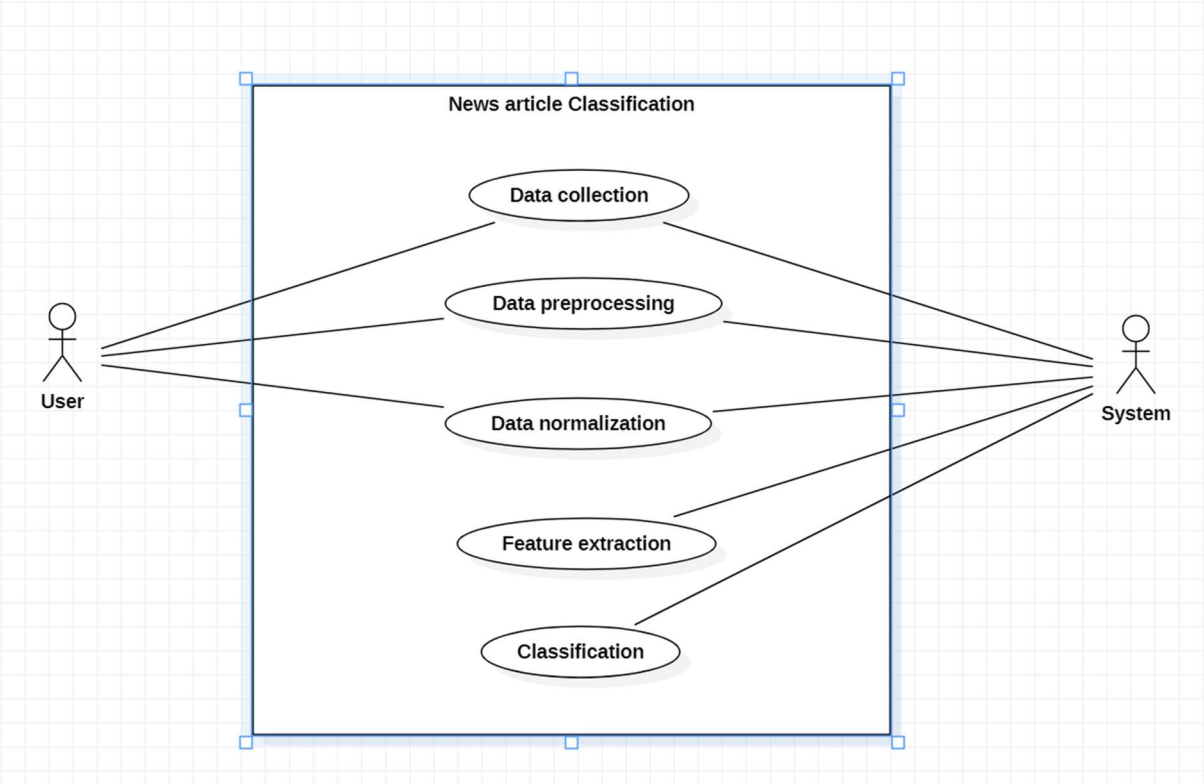
CHAPTER - 4
SYSTEM IMPLEMENTATION

UML Diagrams:

The UML stands for Unified Modeling Language. It is more than simply a notation for drawing diagrams; it is also a full language for capturing (semantics) and conveying (syntax) information about a subject for the purpose of communication.

Use Case Diagram:

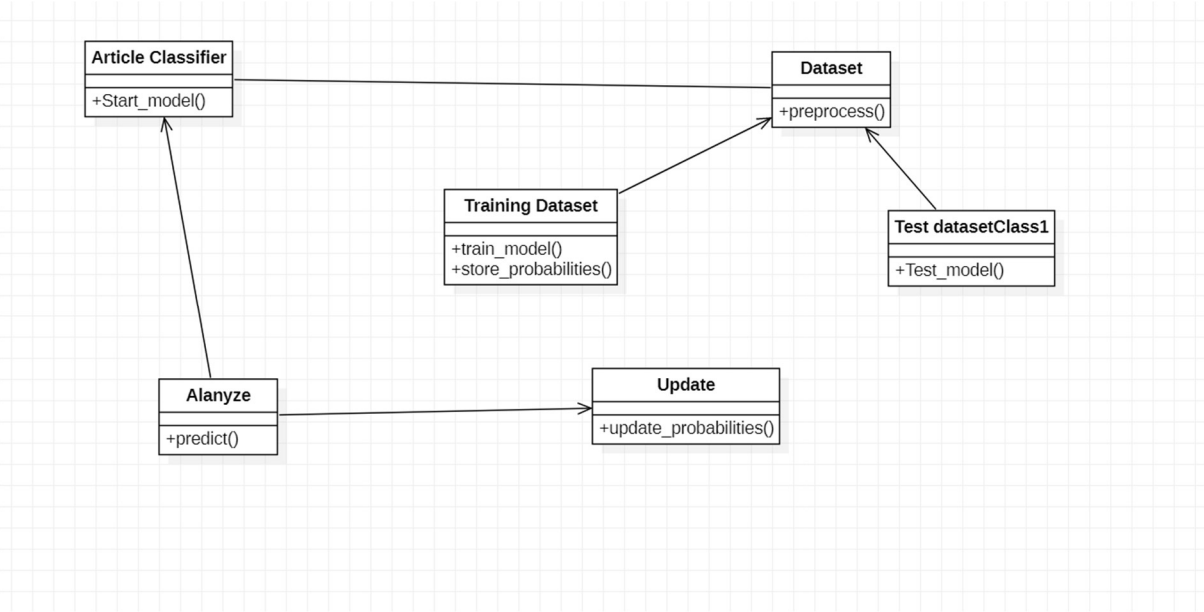
Use case diagrams are used to depict the system's dynamic nature. We need certain parameters that will interact with the use case diagrams because they are dynamic. The agents are referred to as "actors." Use case diagrams are used to model an application's system or subsystem. A single use case diagram depicts a system's specific capabilities.



Use case Diagram

Class Diagram:

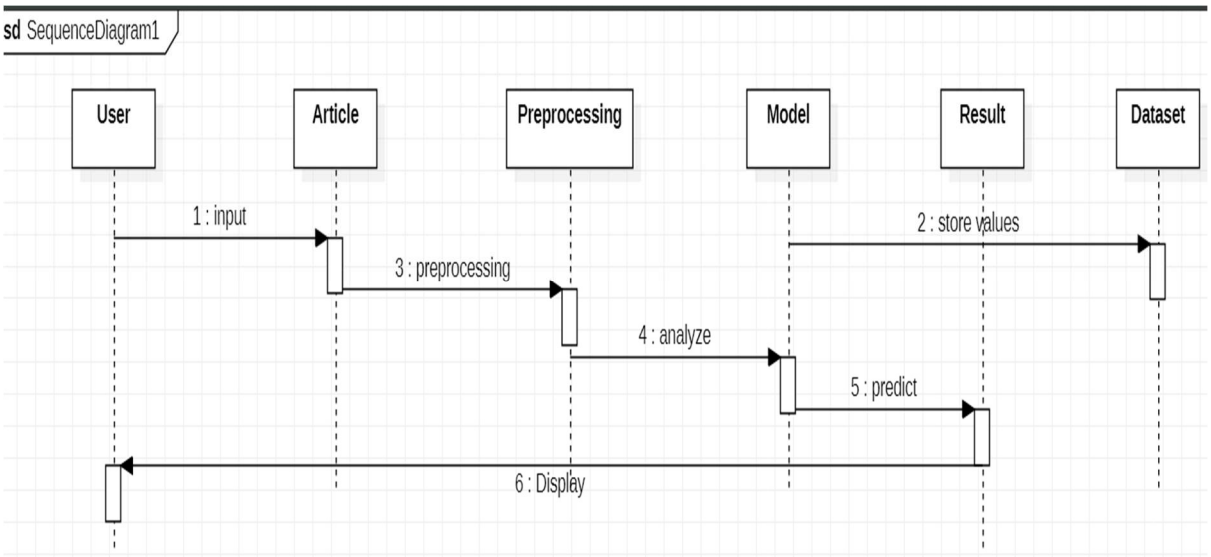
A class diagram is a representation of a group of objects with similar features, actions, relationships, and meanings. The diagram is static. The class diagram depicts the system's limitations as well as the properties and activities of a class. A collection of classes, interfaces, affiliations, collaborations, and restrictions are shown in the class diagram. The structural diagram is also known as the class diagram.



Class Diagram

Sequence Diagram:

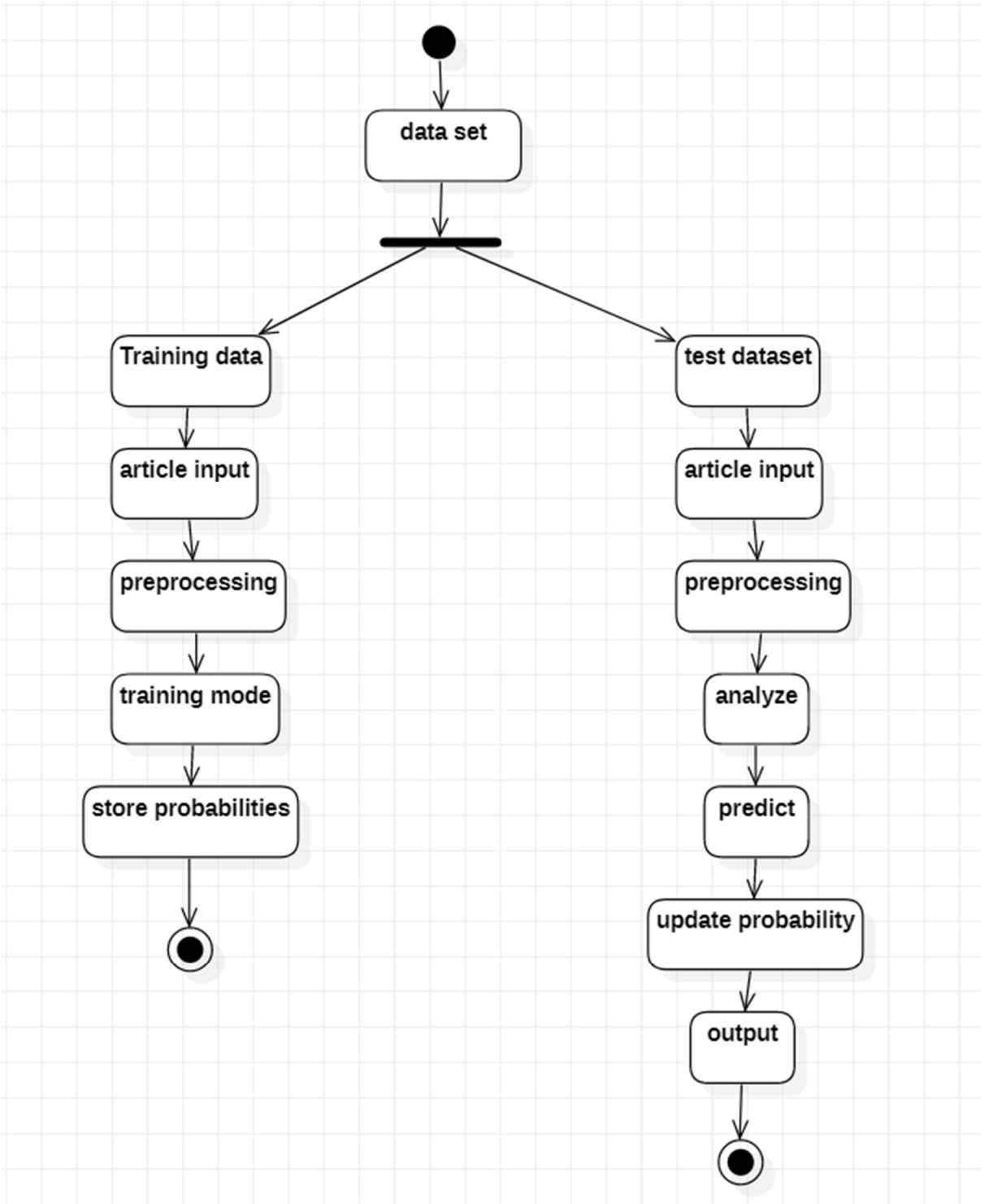
It is a diagram of interaction. It depicts how things in a series interact with one another. The time sequence of communications is shown in sequence diagrams. They are used to capture a system's dynamic behavior.



Sequence Diagram

Activity Diagram:

Action diagrams are flow chart diagrams that depict the system's movement from one activity to the next. The operation of the system is characterized as a system activity. As a result, the control flow is transferred from one operation to the next. This flow might be branching, sequential, or concurrent. Using features like as fork, join, and others, activity diagrams deal with all types of flow.



Activity Diagram

IMPLEMENTATION:

DATA PREPROCESSING:

```
import json

filepath =  'News_Category_Dataset_v3.json'
with open(filepath, encoding="utf-8") as json_lines_file:
    data = []
    for line in json_lines_file:
        data.append(json.loads(line))
```

Downloaded data is of the ‘json’ form. Json file contains unicode characters which are interpreted as ascii characters. So the json file is changed to csv file using ‘utf-8’ encoding.

```
file = open("data.csv", "w")

for i in range(len(data)):
    if data[i]['category'] in required_category:
        short_description = data[i]['short_description'].encode('ascii', 'ignore').decode('ascii')
        category = data[i]['category'].encode('ascii', 'ignore').decode('ascii')
        file.write(str(i + 1) + ',' + short_description + ',' + category + '\n')
```

A new data.csv file is created and data is written to the new file.

DATA ANALYSIS:

```
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt

#Load Data
train_data = pd.read_csv('../DATA/data.csv')
```

Importing pandas, numpy, math and matplotlib libraries. Loading data from data.csv file using pandas and converting it to dataframe.

```
train_data.head()
```

	ArticleId	Text	Category
0	14	One man's claims that he scammed people on the...	TECH
1	18	Maury Wills who helped the Los Angeles Dodgers...	SPORTS
2	21	For the past 18 months Hollywood has effective...	ENTERTAINMENT
3	22	President issues vow as tensions with China rise.	POLITICS
4	25	An annual celebration took on a different feel...	POLITICS

Printing head of the dataframe to check the columns of the data. It provides the basic understanding of the data and their data types.

```
# Basic Data Exploration
print("NUMBER OF DATA POINTS -",train_data.shape[0])
print("NUMBER OF FEATURES -",train_data.shape[1])
print("FEATURES -",train_data.columns.values)
```

```
NUMBER OF DATA POINTS - 64351
NUMBER OF FEATURES - 3
FEATURES - ['ArticleId' 'Text' 'Category']
```

Printing number of rows, number of columns and features of the dataframe. Number of rows gives us the total data points. Number of columns gives the features of the dataframe.

```
# Data Points per Category
train_data['Category'].value_counts()
```

```
POLITICS         34696
ENTERTAINMENT    16953
BUSINESS          5753
SPORTS            4933
TECH              1958
Name: Category, dtype: int64
```

Printing total number of features of each category. This helps to normalize the data according to the algorithm needs.

```
# Data Cleaning. Removing rows with missing values
train_data.dropna(inplace=True)
train_data.isna().sum()
```

```
ArticleId    0
Text         0
Category     0
dtype: int64
```

Cleaning data is one of the important steps in data analysis. Here we are deleting rows from the data frame which have no data data or null value.

```
target_category = train_data['Category'].unique()
print(target_category)
```

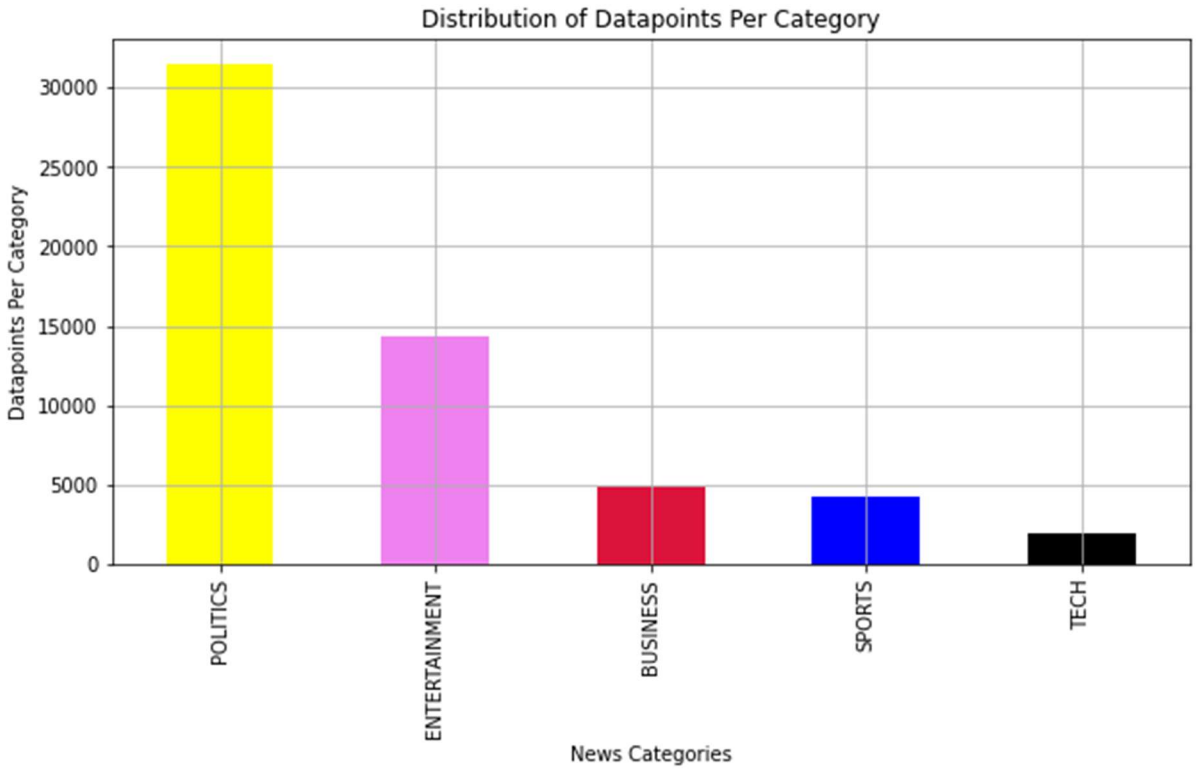
```
['TECH' 'SPORTS' 'ENTERTAINMENT' 'POLITICS' 'BUSINESS']
```

Printing all unique values of the category. This helps to find the number of clusters or number of classes in classification problems.

```
# Plotting data points per category
news_cat = train_data['Category'].value_counts()

plt.figure(figsize=(10,5))
my_colors = ['yellow','violet','crimson','blue','black']
news_cat.plot(kind='bar', color=my_colors)
plt.grid()
plt.xlabel("News Categories")
plt.ylabel("Datapoints Per Category")
plt.title("Distribution of Datapoints Per Category")
plt.show()
```

A picture is worth a thousand words. So, here is a bar graph of Category Value vs Data Points Per Category.



This bar graph concludes that the category with politics has the highest number of datapoints, while the tech category has the least number of datapoints.

DATA VISUALIZATION:

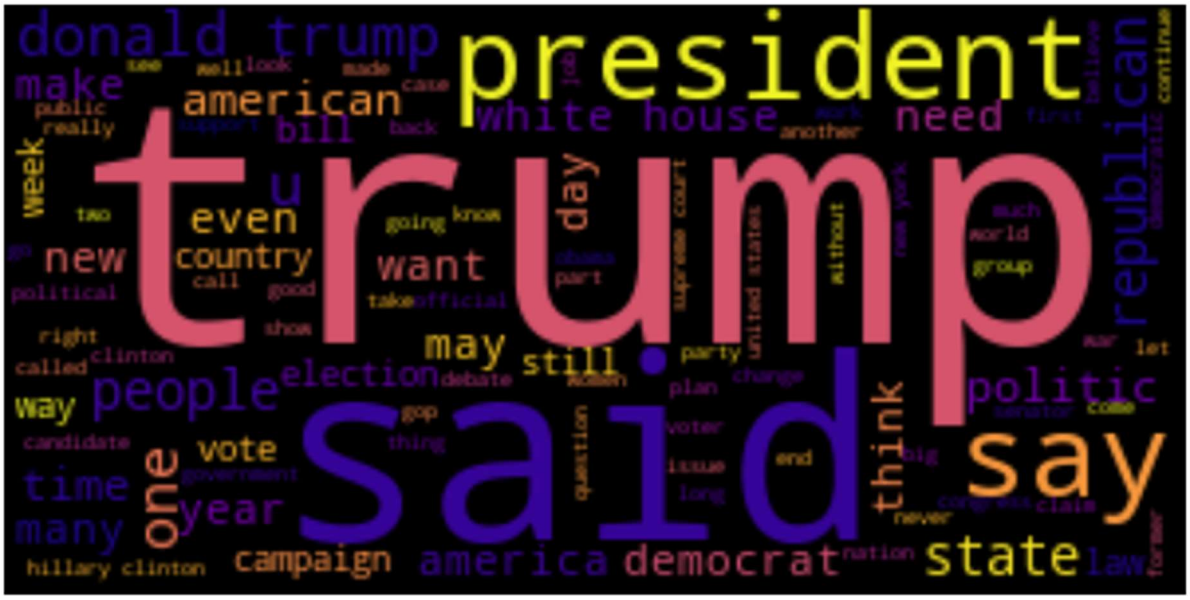
Word Clouds are the one of the next ways to visualize the textual data. Here are the generated word clouds for the data.

WORD CLOUD FOR POLITICS

```
plt.figure(figsize=(12,12))
wc = WordCloud(max_words=100,
               min_font_size=5,
               height=150,
               width=300,
               background_color='black',
               contour_color='black',
               colormap='plasma',
               repeat=False,
               stopwords=STOPWORDS).generate(' '.join([str(item) for item in politics]))

plt.title("Wordcloud for Politics", size=15, weight='bold')
plt.imshow(wc, interpolation= "bilinear")
plt.axis('off')
```

Wordcloud for Politics

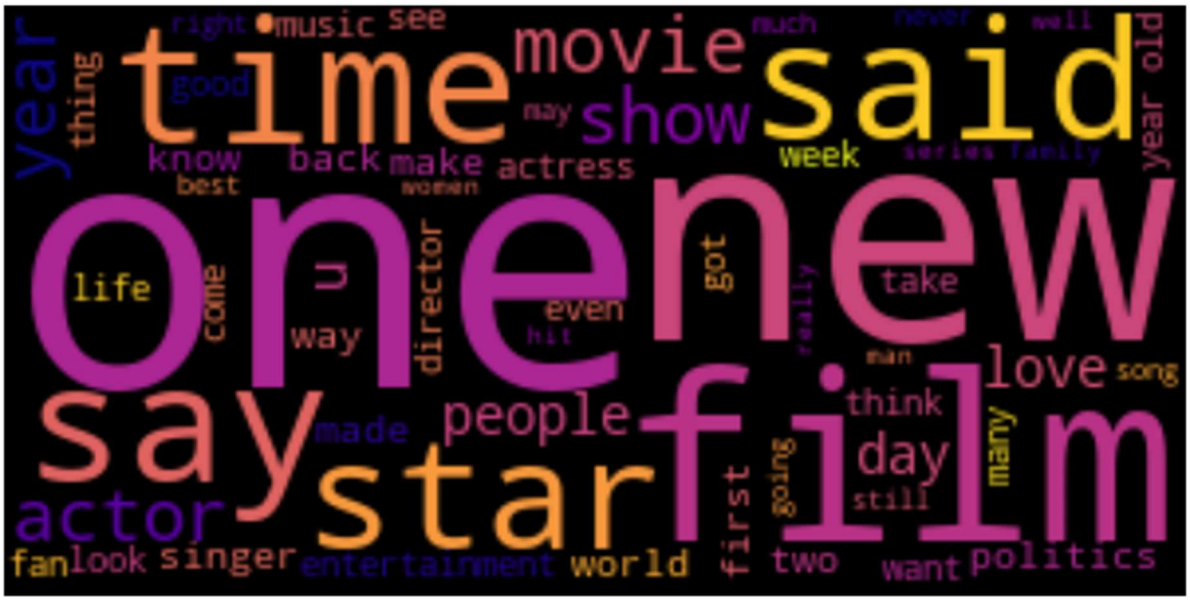


WORD CLOUD FOR ENTERTAINMENT

```
plt.figure(figsize=(12,12))
wc = WordCloud(max_words=100,
               min_font_size=6,
               height=150,
               width=300,
               background_color='black',
               contour_color='black',
               colormap='plasma',
               repeat=False,
               stopwords=STOPWORDS).generate(' '.join([str(item) for item in entertainment]))

plt.title("Wordcloud for Entertainment", size=15, weight='bold')
plt.imshow(wc, interpolation= "bilinear")
plt.axis('off')
```

Wordcloud for Entertainment

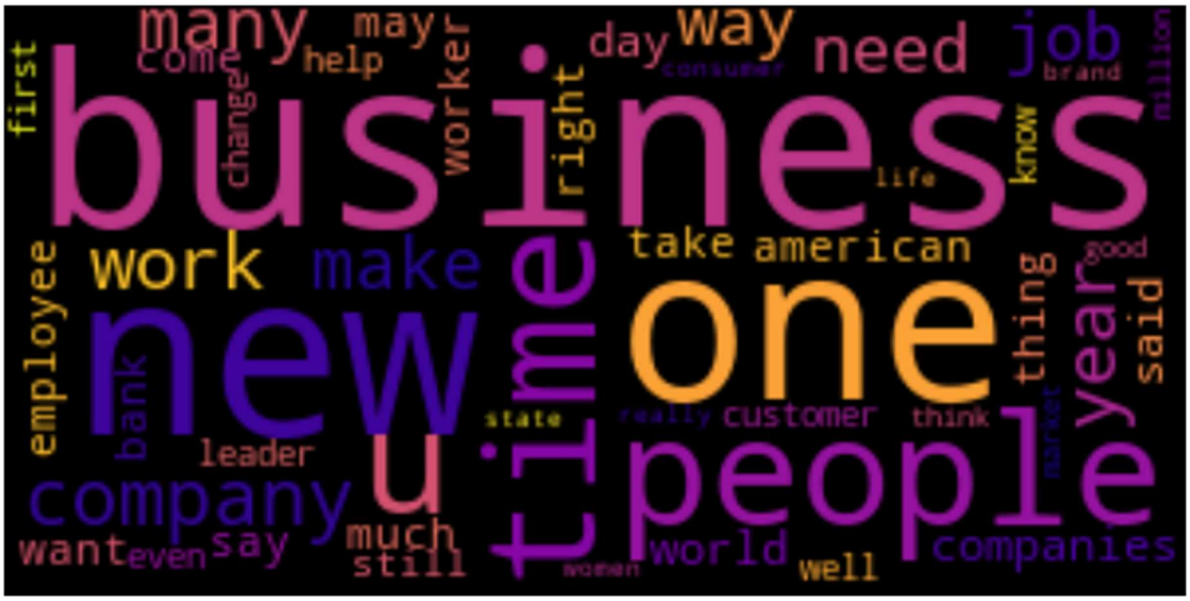


WORD CLOUD FOR BUSINESS

```
plt.figure(figsize=(12,12))
wc = WordCloud(max_words=100,
               min_font_size=6,
               height=150,
               width=300,
               background_color='black',
               contour_color='black',
               colormap='plasma',
               repeat=False,
               stopwords=STOPWORDS).generate(' '.join([str(item) for item in business]))

plt.title("Wordcloud for Business", size=15, weight='bold')
plt.imshow(wc, interpolation= "bilinear")
plt.axis('off')
```

Wordcloud for Business



WORD CLOUD FOR SPORTS

```
plt.figure(figsize=(12,12))
wc = WordCloud(max_words=100,
               min_font_size=6,
               height=150,
               width=300,
               background_color='black',
               contour_color='black',
               colormap='plasma',
               repeat=False,
               stopwords=STOPWORDS).generate(' '.join([str(item) for item in sports]))

plt.title("Wordcloud for Sports", size=15, weight='bold')
plt.imshow(wc, interpolation= "bilinear")
plt.axis('off')
```

Wordcloud for Sports

WORD CLOUD FOR TECH

```
plt.figure(figsize=(12,12))
wc = WordCloud(max_words=100,
               min_font_size=6,
               height=150,
               width=300,
               background_color='black',
               contour_color='black',
               colormap='plasma',
               repeat=False,
               stopwords=STOPWORDS).generate(' '.join([str(item) for item in tech]))

plt.title("Wordcloud for Tech", size=15, weight='bold')
plt.imshow(wc, interpolation= "bilinear")
plt.axis('off')
```

DATA CLEANING:

```
import nltk
from nltk.corpus import stopwords
print(stopwords.words('english'))
```

✓ 40.7s Python

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours',
'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its',
'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these',
'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing',
'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about',
'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',
'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any',
'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too',
'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y',
'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't",
'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't",
'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Textual data contains different types of words with different parts of speech and their usage. The English language contains stopwords. Stopwords are the set of common words in the language which doesn't provide any syntactical meaning. These are eliminated, allowing machine learning applications to focus on the important words.

```
import warnings
warnings.filterwarnings("ignore")
from nltk.corpus import stopwords
import nltk
import re

# Loading stopwords from nltk library
stop_words = set(stopwords.words('english'))
# Function for text preprocessing
def txt_preprocessing(total_text, index, column, df):
    if type(total_text) is not int:
        string = ""
        # Replace every special character with space
        total_text = re.sub('[^a-zA-Z0-9\n]', ' ', total_text)
        # Remove multiple spaces
        total_text = re.sub('\s+',' ', total_text)
        # Converting to lowercase
        total_text = total_text.lower()

        for word in total_text.split():
        # If word is not a stopword then retain that word from the data
            if not word in stop_words:
                string += word + " "
        df[column][index] = string
```

Special characters and unrequired spaces are removed from the strings. All words in the string are converted to lowercase and stop words are removed from the strings.

```
# Preprocessing the data

for index, row in train_data.iterrows():
    if type(row['Text']) is str:
        txt_preprocessing(row['Text'], index, 'Text', train_data)

train_data.head()
```

	ArticleId	Text	Category
0	14	one man claims scammed people platform caused ...	TECH
1	18	maury wills helped los angeles dodgers win thr...	SPORTS
2	21	past 18 months hollywood effectively boycotted...	ENTERTAINMENT
3	22	president issues vow tensions china rise	POLITICS
4	25	annual celebration took different feel russia ...	POLITICS

DataFrame is printed after cleaning and removing stop words from the data.

DATA DISTRIBUTION:

```
# Dividing the data into train and test set
from sklearn.model_selection import train_test_split
X_train = train_data
y_train = train_data['Category']

X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.20, stratify=y_train, random_state=0)

print("NUMBER OF DATA POINTS IN TRAIN DATA :", X_train.shape[0])
print("NUMBER OF DATA POINTS IN CROSS VALIDATION DATA :", X_cv.shape[0])
```

Available data is distributed as training data and test data with the help of train_test_split from sklearn model_selection.

```
NUMBER OF DATA POINTS IN TRAIN DATA : 45578
NUMBER OF DATA POINTS IN CROSS VALIDATION DATA : 11395
```

Number of data points in training data and cross validation data is printed.

```
from sklearn.feature_extraction.text import CountVectorizer

text_vectorizer = CountVectorizer(min_df=3)
train_text = text_vectorizer.fit_transform(X_train['Text'])

# Getting all the feature names
train_text_features = text_vectorizer.get_feature_names()
train_text_fea_counts = train_text.sum(axis=0).A1

text_fea_dict = dict(zip(list(train_text_features),train_text_fea_counts))
```

Machines cannot understand characters and words. So, when dealing with text data we need to represent it in numbers to be understood by the machine. Countvectorizer is a method to convert text to numerical data.

```
print("Total Number of Unique Words in Train Data :",len(train_text_features))
```

```
Total Number of Unique Words in Train Data : 15103
```

Printing the total number of unique words in the train data after the count vectorizer.

```
print(text_fea_dict)
```

```
'abstract': 3, 'absurd': 17, 'abu': 5, 'abundance': 9, 'abundant': 3, 'abuse': 107, 'abused': 11, 'abuser': 3, 'abusers': 6,
'abuses': 15, 'abusing': 14, 'abusive': 10, 'abysmal': 3, 'abyss': 5, 'aca': 29, 'academic': 13, 'academics': 13, 'academy': 63,
'acc': 3, 'accelerate': 6, 'accelerated': 5, 'accelerating': 4, 'accent': 4, 'accept': 54, 'acceptable': 19, 'acceptance': 15,
'accepted': 21, 'accepting': 16, 'accepts': 10, 'access': 161, 'accessed': 3, 'accessible': 13, 'accessories': 4, 'accessory':
3, 'accident': 19, 'accidental': 3, 'accidentally': 9, 'accidents': 4, 'accio': 3, 'acclaimed': 18, 'accolades': 4,
```

Printing the dictionary of text features with their count values.

```
from sklearn.preprocessing import normalize

train_text_ohe = normalize(train_text_ohe, axis=0)
cv_text_ohe = text_vectorizer.transform(X_cv['Text'])
cv_text_ohe = normalize(cv_text_ohe, axis=0)
```

Data is normalized using sklearn pre-processing. After normalizing data points in all categories are near to equal representing almost equal percentage.

REFERENCES

1. https://www.ijitee.org/wp-content/uploads/papers/v9i5/E2753039520.pdf
2. https://file.techscience.com/ueditor/files/cmc/TSP_CMC-71-1/TSP_CMC_22011/TSP_CMC_22011.pdf
3. https://www.researchgate.net/publication/220195844_An_alternative_approach_for_statistical_single-label_document_classification_of_newspaper_articles

B3-1

ORIGINALITY REPORT

20%	12%	7%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijitee.org Internet Source	2%
2	Submitted to Technological Institute of the Philippines Student Paper	1%
3	towardsdatascience.com Internet Source	1%
4	Submitted to University of North Texas Student Paper	1%
5	Submitted to The NorthCap University, Gurugram Student Paper	1%
6	datascience.ase.ro Internet Source	1%
7	assignmentoverflow.com Internet Source	1%
8	huggingface.co Internet Source	1%

9	Submitted to Letterkenny Institute of Technology Student Paper	1%
10	Submitted to University of Sheffield Student Paper	1%
11	Submitted to Southern New Hampshire University - Continuing Education Student Paper	1%
12	Submitted to Milwaukee School of Engineering Student Paper	1%
13	Submitted to University of Hertfordshire Student Paper	1%
14	Submitted to North West University Student Paper	1%
15	www.coursehero.com Internet Source	1%
16	Submitted to Lithan Academy Pte Ltd Student Paper	1%
17	web.archive.org Internet Source	<1%
18	machinelearningmastery.com Internet Source	<1%
19	Submitted to Asia Pacific International College Student Paper	<1%

20	Submitted to The Manchester College Student Paper	<1%
21	nc.mmudev.com Internet Source	<1%
22	Jia-Xin Liu. "Research of reading content recommendation based on behavior mining", 2008 International Conference on Information and Automation, 06/2008 Publication	<1%
23	Lecture Notes in Business Information Processing, 2014. Publication	<1%
24	P. Caudal, E. Simonetto, V. Merrien-Soukatchoff, T. J. B. Dewez. "SEMI-AUTOMATIC ROCK MASS GEOMETRY ANALYSIS FROM A DENSE 3D POINT CLOUD WITH DISCONTINUITYLAB", ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2020 Publication	<1%
25	dokumen.pub Internet Source	<1%
26	"Newspaper Article Classification using Machine Learning Techniques", International Journal of Innovative Technology and Exploring Engineering, 2020 Publication	<1%

27	"Web Data Mining", Springer Science and Business Media LLC, 2007 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		