GROUP NUMBER: 09

- Venuja Udukumbura - 226123A
- Tharusha Perera - 226092B
- Minsara Bandara - 226013L
- Dineth Akash - 226003G
- Saveen Fernando - 226030K

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Task 1: Preparing Data

Define File Paths

```
file_paths = {
    "Business": "/content/drive/MyDrive/Business.xlsx",
    "Opinion": "/content/drive/MyDrive/Opinion.xlsx",
    "Political_gossip":
"/content/drive/MyDrive/Political_gossip.xlsx",
    "Sports": "/content/drive/MyDrive/Sports.xlsx",
    "World_news": "/content/drive/MyDrive/World_news.xlsx"
}
```

Read and Process Each File

```
import pandas as pd

dfs = []

for category, file in file_paths.items():
    df = pd.read_excel(file)
    df["class"] = category
    if "title" in df.columns:
        df.drop(columns=["title"], inplace=True)  # Drop the 'title'
column
    dfs.append(df)
```

Merge All Data into One DataFrame

```
final_df = pd.concat(dfs, ignore_index=True)
```

Remove Duplicate Rows

```
final_df.drop_duplicates(subset=["content"], inplace=True)

#checking a sample
final_df.sample(10)
```


```
{"summary":"{\n  \"name\": \"final_df\",\n  \"rows\": 10,\n
\"fields\": [\n    {\n      \"column\": \"Unnamed: 0\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
52,\n        \"min\": 20,\n        \"max\": 208,\n
\"num_unique_values\": 10,\n        \"samples\": [\n          159,\n
20,\n          124\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"content\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 10,\n        \"samples\": [\n
\"India and China have reached a consensus over resuming the Indian
pilgrim\\u2019s pilgrimage Kailash Mansarovar Yatra after years of
border standoff. Both sides have also emphasised on the measures to
promote and maintain peace with special mention to cross-border river
cooperation and Nathula border trade.\",\n          \"The Central Bank
of Sri Lanka has announced a T-Bill auction totaling LKR 115.0Bn,
scheduled for 19th Feb-25. CBSL aims to raise LKR 25.0Bn from 3M, LKR
60.0Bn from 6M, and LKR 30.0Bn from 12M T-Bills respectively. Today,
the secondary market experienced a volatile trading session, marked by
strong buying interest among market participants. As a result,
secondary market trades witnessed high trading volumes, however the
yield curve remained broad\",\n          \"As the poverty rates in the
country declined below 5 per cent in 2024, a research study by State
Bank of India (SBI) also highlighted that the extreme poverty in the
country has reduced to minimal.\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"class\",\n      \"properties\": {\
n        \"dtype\": \"string\",\n        \"num_unique_values\": 5,\n
\"samples\": [\n          \"Business\",\n          \"World_news\",\n
\"Political_gossip\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


Save the Final Dataset

```
final_file_path = "/content/drive/MyDrive/Daily_Mirror_News.xlsx"
final_df.to_excel(final_file_path, index=False)

print("Final dataset saved at:", final_file_path)
```

```
Final dataset saved at: /content/drive/MyDrive/Daily_Mirror_News.xlsx
```

```
final_df.head(10)
```


```
{"summary":"{\n  \"name\": \"final_df\",\n  \"rows\": 1016,\n
\"fields\": [\n    {\n      \"column\": \"Unnamed: 0\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

```
59,\n        \"min\": 0,\n        \"max\": 209,\n
\"num_unique_values\": 210,\n        \"samples\": [\n          30,\n
173,\n          84\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"content\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 1015,\n        \"samples\": [\n
\"The Sri Lanka Netball Team will leave for India today to participate
in the 13th Asian Senior Netball Championship 2024, which will be held
from October 18 to 27 at the Koramangala Indoor Stadium in
Bengaluru.\",\n          \"An informal decision is said to have been
taken to keep away a certain prominent Diyawanna member from all
government meetings and other collective activity.\",\n          \"The
first part of this article published last week briefly outlined the
history of Sri Lanka\\u2019s national anthem being sung in Tamil. In
this second and final part, the focus would be on the regress and
progress of the national anthem rendition in Tamil over the years. As
stated earlier the Tamil national anthem issue is in a sense
symptomatic of the escalating ethnic crisis in Sri Lanka.\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"class\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 5,\n        \"samples\": [\n
\"Opinion\",\n          \"World_news\",\n
\"Political_gossip\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"final_df"}
```


Task 2: EDA & Text Preprocessing

Import Libraries and Load Data

```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from collections import Counter
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
import re

# Download necessary NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
# Load dataset
df = pd.read_excel("/content/drive/MyDrive/Daily_Mirror_News.xlsx")


[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...

# Display basic info
print(df.info())
print(df.head())

# Check for missing values
print("Missing values per column:")
print(df.isnull().sum())

# Drop missing values
df.dropna(inplace=True)

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1016 entries, 0 to 1015
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  1016 non-null   int64
 1   content     1015 non-null   object
 2   class       1016 non-null   object
dtypes: int64(1), object(2)
memory usage: 23.9+ KB
None
   Unnamed: 0                                            content
class
0           0  Sri Lanka’s inflation is expected to increase ...
Business
1           1                  President Anura Kumara Dissanayake
Business
2           2  As artificial intelligence (AI) evolves from b...
Business
3           3  A group of Ride for Ceylon participants from t...
Business
4           4  The ASPI closed in green as a result of price ...
Business
Missing values per column:
Unnamed: 0    0
content       1
class         0
dtype: int64
```

Word Cloud

```
from wordcloud import WordCloud

# Function to generate a word cloud
def generate_wordcloud(text, title):
    wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(" ".join(text))
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.title(title, fontsize=14)
    plt.show()

# Generate word cloud for entire dataset
generate_wordcloud(df['content'], "Word Cloud for News Articles")
```

Word Cloud for Each Class

```
# Get unique classes
categories = df["class"].unique()

# Create a 2x2 subplot layout
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

axes = axes.flatten()

# Generate a word cloud for each category (max 4)
for i, category in enumerate(categories[:4]):
```

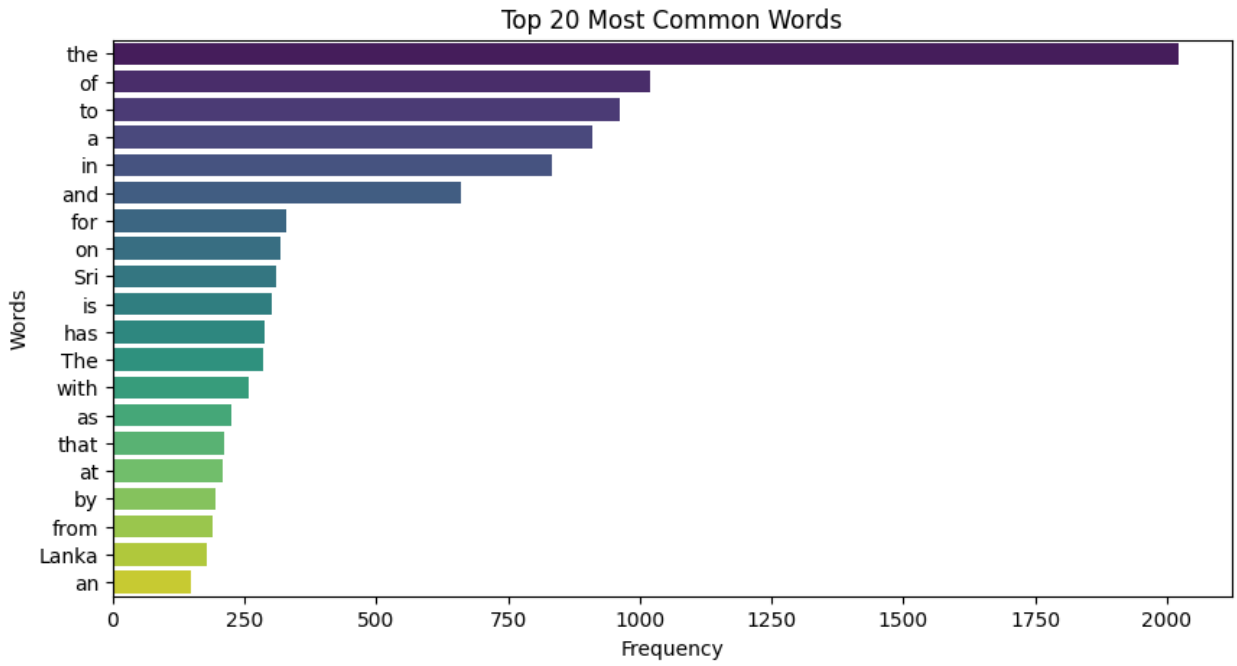
```
    category_text = df[df["class"] == category]["content"]

    # Generate word cloud
    wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(" ".join(category_text))

    # Plot the word cloud
    axes[i].imshow(wordcloud, interpolation='bilinear')
    axes[i].axis("off")
    axes[i].set_title(f"Word Cloud for {category}")

# Adjust layout
plt.tight_layout()
plt.show()
```

N-gram Analysis

```
import seaborn as sns

# Function to plot most common words
def plot_top_n_words(text, n=20):
    words = " ".join(text).split()
    common_words = Counter(words).most_common(n)
    df_words = pd.DataFrame(common_words, columns=['Word',
'Frequency'])
```

```
    plt.figure(figsize=(10,5))
    sns.barplot(x='Frequency', y='Word', data=df_words,
palette='viridis')
    plt.xlabel("Frequency")
    plt.ylabel("Words")
    plt.title(f"Top {n} Most Common Words")
    plt.show()

# Plot top 20 words
plot_top_n_words(df['content'])
```

```
<ipython-input-13-3a93204cbc41>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='Frequency', y='Word', data=df_words,
palette='viridis')
```

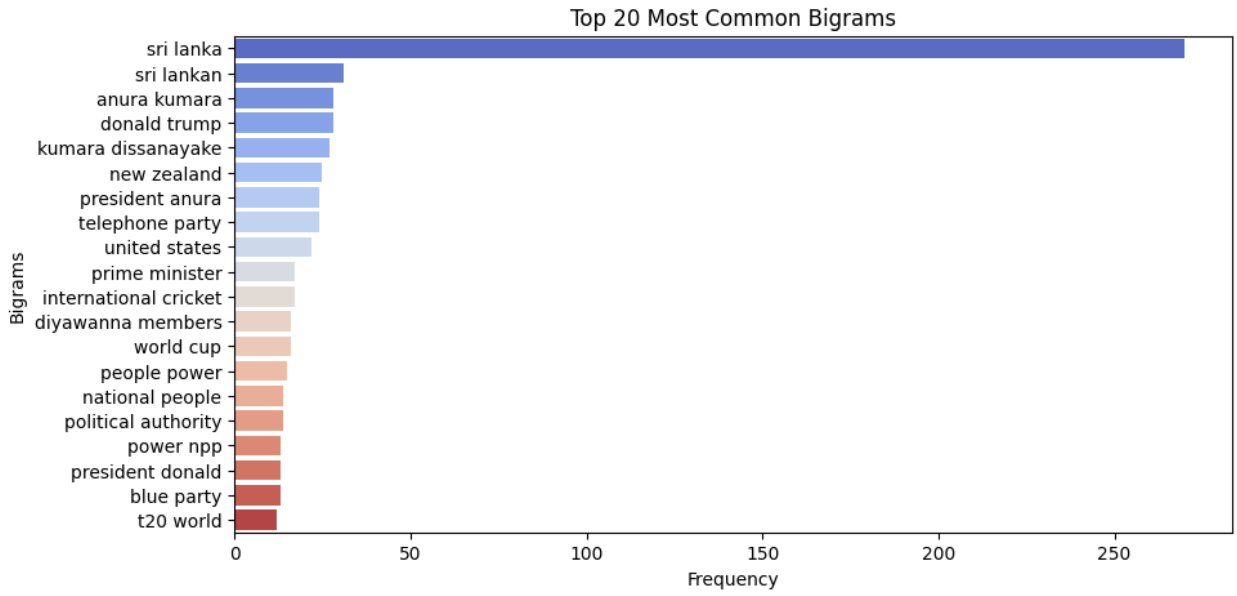


Most Common Bigrams

```
from sklearn.feature_extraction.text import CountVectorizer
import seaborn as sns

def get_top_n_bigrams(corpus, n=20):
    vectorizer = CountVectorizer(ngram_range=(2,2),
stop_words='english')
    X = vectorizer.fit_transform(corpus)
```

```
    bigram_freq = X.toarray().sum(axis=0)
    bigram_dict = {bigram: bigram_freq[idx] for bigram, idx in
vectorizer.vocabulary_.items()}
    return Counter(bigram_dict).most_common(n)

# Get top 20 bigrams
bigrams = get_top_n_bigrams(df['content'])

# Convert to DataFrame for plotting
bigram_df = pd.DataFrame(bigrams, columns=['Bigram', 'Frequency'])

# Plot bigram frequency
plt.figure(figsize=(10, 5))
sns.barplot(y=bigram_df["Bigram"], x=bigram_df["Frequency"],
palette="coolwarm")
plt.xlabel("Frequency")
plt.ylabel("Bigrams")
plt.title("Top 20 Most Common Bigrams")
plt.show()
```

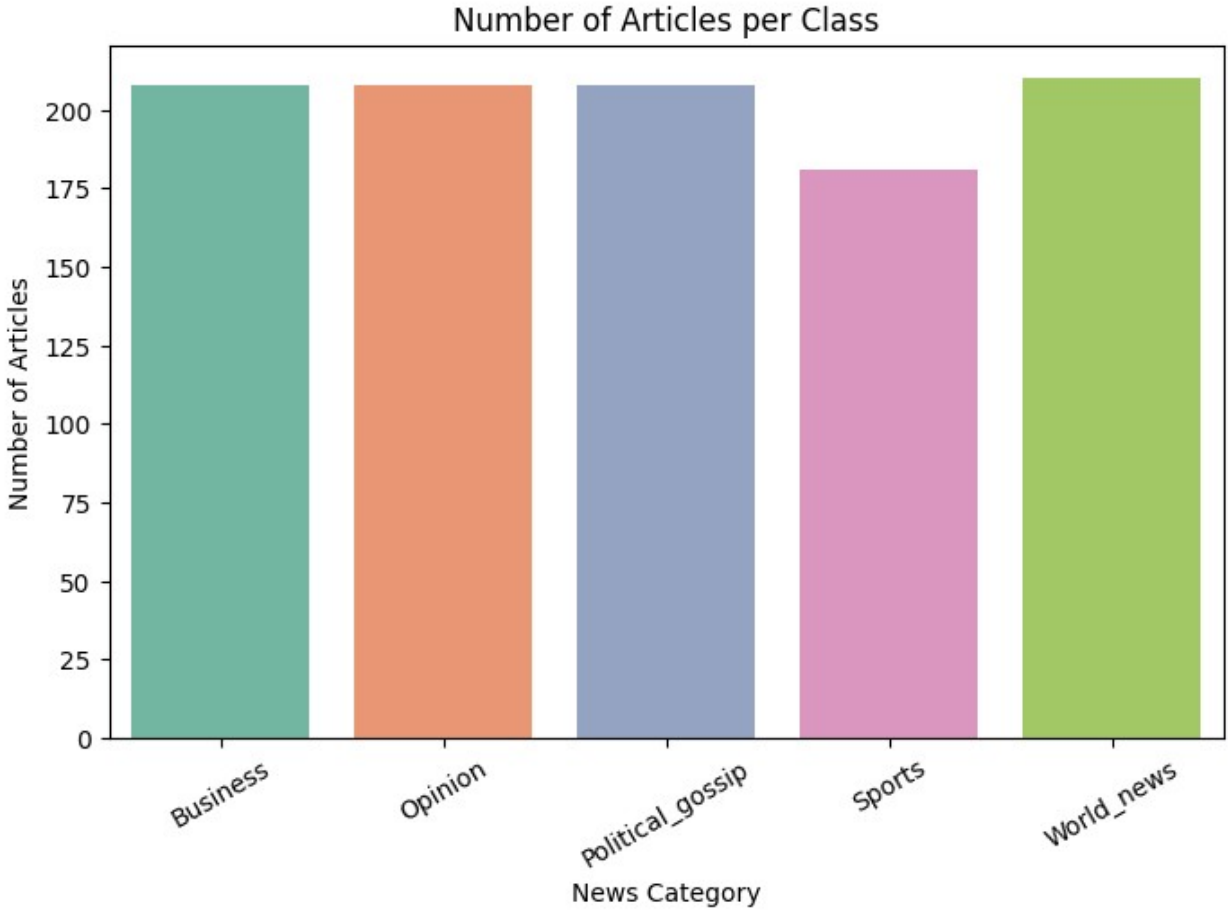
```
<ipython-input-14-35f04ea1d762>:19: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(y=bigram_df["Bigram"], x=bigram_df["Frequency"],
palette="coolwarm")
```



Bar Chart for Number of Articles per Class

```
plt.figure(figsize=(8, 5))
sns.countplot(x=df["class"], palette="Set2")
plt.xlabel("News Category")
plt.ylabel("Number of Articles")
plt.title("Number of Articles per Class")
plt.xticks(rotation=30)
plt.show()
```

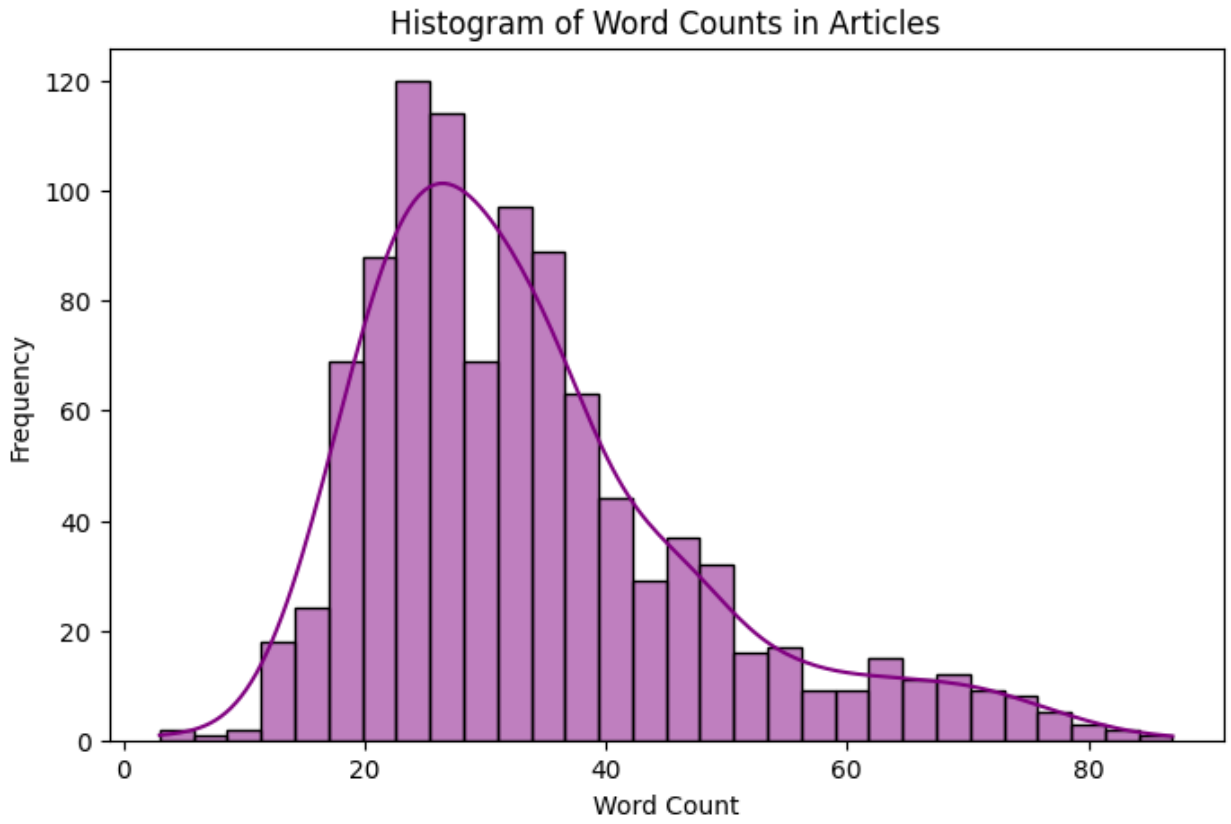
```
<ipython-input-15-fe24c2e979f7>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(x=df["class"], palette="Set2")
```



Histogram of Word Counts

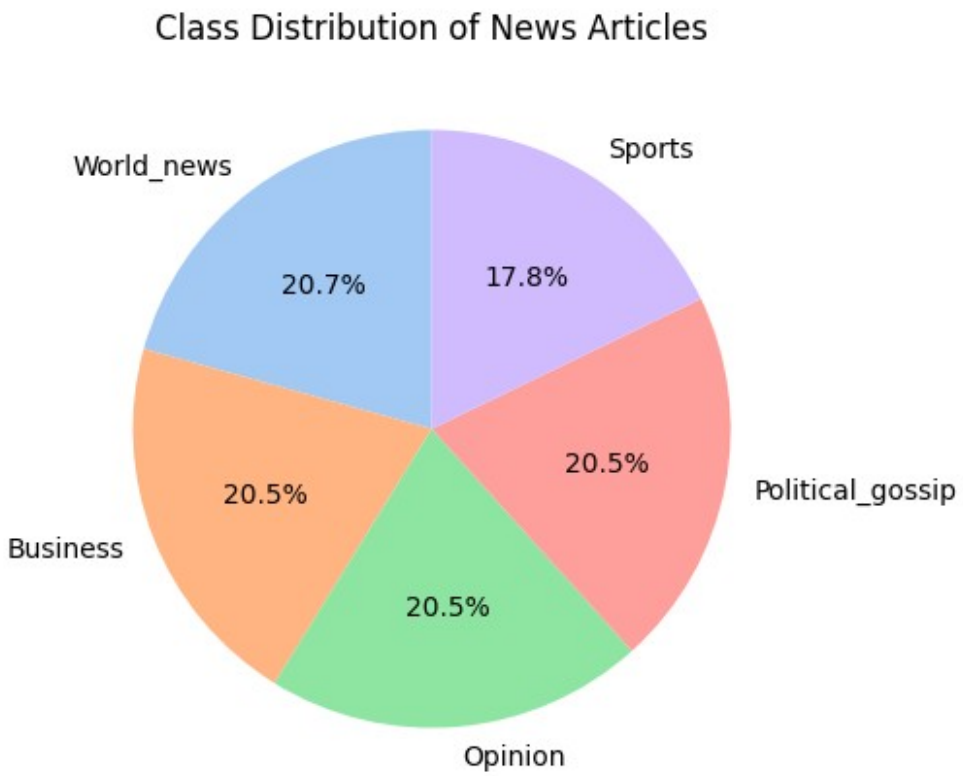
```
df["word_count"] = df["content"].apply(lambda x: len(str(x).split()))

plt.figure(figsize=(8, 5))
```

```
sns.histplot(df["word_count"], bins=30, kde=True, color="purple")
plt.xlabel("Word Count")
plt.ylabel("Frequency")
plt.title("Histogram of Word Counts in Articles")
plt.show()
```



Pie Chart for Class Distribution

```
plt.figure(figsize=(8, 5))
df["class"].value_counts().plot.pie(
    autopct="%1.1f%%",
    colors=sns.color_palette("pastel"),
    startangle=90
)
plt.ylabel("")
plt.title("Class Distribution of News Articles")
plt.show()
```



Initialize Preprocessing Components

```
# Define stopwords, stemmer, and lemmatizer
stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()
```

Define Preprocessing Function

```
def preprocess_text(text):
    steps = {}

    # Original text
    steps["Original"] = text

    # Lowercasing
    text = text.lower()
    steps["Lowercase"] = text

    # Removing special characters and numbers
    text = re.sub(r'[^a-zA-Z]', ' ', text)
    steps["No Special Characters"] = text

    # Tokenization
    tokens = word_tokenize(text)
```

```
    steps["Tokens"] = tokens

    # Removing stopwords
    filtered_tokens = [word for word in tokens if word not in
stop_words]
    steps["No Stopwords"] = filtered_tokens

    # Stemming
    stemmed_tokens = [stemmer.stem(word) for word in filtered_tokens]
    steps["Stemmed"] = stemmed_tokens

    # Lemmatization
    lemmatized_tokens = [lemmatizer.lemmatize(word) for word in
stemmed_tokens]
    steps["Lemmatized"] = lemmatized_tokens

    # Join back to text
    preprocessed_text = " ".join(lemmatized_tokens)
    steps["Final_Text"] = preprocessed_text

    return steps


import nltk
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

```
True
```

Apply Preprocessing on a Sample Record

```
# Select a sample of the dataset (we got the first 5 rows)
sample_text = df['content'].head(5)

# Apply the preprocessing function on the sample
preprocessed_sample = sample_text.apply(preprocess_text)

# Print the preprocessed sample to inspect
for idx, text_steps in preprocessed_sample.items():
    print(f"Sample {idx}:")
    for step, text in text_steps.items():
        print(f"{step}: {text}")
    print("\n" + "="*50 + "\n")
```

```
Sample 0:
Original: Sri Lanka’s inflation is expected to increase “sizeably” in
the third quarter this year (3Q25), with the possibility of it
```

```
hovering around 2 percentage points above the inflation target in mid-2026, the Central Bank of Sri Lanka (CBSL) said in its monetary policy report that was released yesterday.
Lowercase: sri lanka’s inflation is expected to increase “sizeably” in the third quarter this year (3q25), with the possibility of it hovering around 2 percentage points above the inflation target in mid-2026, the central bank of sri lanka (cbsl) said in its monetary policy report that was released yesterday.
No Special Characters: sri lanka s inflation is expected to increase sizeably  in the third quarter this year   q     with the possibility of it hovering around   percentage points above the inflation target in mid       the central bank of sri lanka  cbsl  said in its monetary policy report that was released yesterday
Tokens: ['sri', 'lanka', 's', 'inflation', 'is', 'expected', 'to', 'increase', 'sizeably', 'in', 'the', 'third', 'quarter', 'this', 'year', 'q', 'with', 'the', 'possibility', 'of', 'it', 'hovering', 'around', 'percentage', 'points', 'above', 'the', 'inflation', 'target', 'in', 'mid', 'the', 'central', 'bank', 'of', 'sri', 'lanka', 'cbsl', 'said', 'in', 'its', 'monetary', 'policy', 'report', 'that', 'was', 'released', 'yesterday']
No Stopwords: ['sri', 'lanka', 'inflation', 'expected', 'increase', 'sizeably', 'third', 'quarter', 'year', 'q', 'possibility', 'hovering', 'around', 'percentage', 'points', 'inflation', 'target', 'mid', 'central', 'bank', 'sri', 'lanka', 'cbsl', 'said', 'monetary', 'policy', 'report', 'released', 'yesterday']
Stemmed: ['sri', 'lanka', 'inflat', 'expect', 'increas', 'sizeabl', 'third', 'quarter', 'year', 'q', 'possibl', 'hover', 'around', 'percentag', 'point', 'inflat', 'target', 'mid', 'central', 'bank', 'sri', 'lanka', 'cbsl', 'said', 'monetari', 'polici', 'report', 'releas', 'yesterday']
Lemmatized: ['sri', 'lanka', 'inflat', 'expect', 'increas', 'sizeabl', 'third', 'quarter', 'year', 'q', 'possibl', 'hover', 'around', 'percentag', 'point', 'inflat', 'target', 'mid', 'central', 'bank', 'sri', 'lanka', 'cbsl', 'said', 'monetari', 'polici', 'report', 'releas', 'yesterday']
Final_Text: sri lanka inflat expect increas sizeabl third quarter year q possibl hover around percentag point inflat target mid central bank sri lanka cbsl said monetari polici report releas yesterday

==================================================

Sample 1:
Original: President Anura Kumara Dissanayake
Lowercase: president anura kumara dissanayake
No Special Characters: president anura kumara dissanayake
Tokens: ['president', 'anura', 'kumara', 'dissanayake']
No Stopwords: ['president', 'anura', 'kumara', 'dissanayake']
Stemmed: ['presid', 'anura', 'kumara', 'dissanayak']
Lemmatized: ['presid', 'anura', 'kumara', 'dissanayak']
```

```
Final_Text: presid anura kumara dissanayak

==================================================

Sample 2:
Original: As artificial intelligence (AI) evolves from being a
technological buzzword to a business imperative, a recent study
commissioned by Microsoft to the International Data Corporation (IDC),
titled ‘The Business Opportunity of AI’, reveals a crucial shift:
businesses are no longer asking if they should integrate AI but rather
how to do so strategically to gain a competitive edge.
Lowercase: as artificial intelligence (ai) evolves from being a
technological buzzword to a business imperative, a recent study
commissioned by microsoft to the international data corporation (idc),
titled ‘the business opportunity of ai’, reveals a crucial shift:
businesses are no longer asking if they should integrate ai but rather
how to do so strategically to gain a competitive edge.
No Special Characters: as artificial intelligence  ai  evolves from
being a technological buzzword to a business imperative  a recent
study commissioned by microsoft to the international data corporation
idc   titled  the business opportunity of ai   reveals a crucial shift
businesses are no longer asking if they should integrate ai but rather
how to do so strategically to gain a competitive edge
Tokens: ['as', 'artificial', 'intelligence', 'ai', 'evolves', 'from',
'being', 'a', 'technological', 'buzzword', 'to', 'a', 'business',
'imperative', 'a', 'recent', 'study', 'commissioned', 'by',
'microsoft', 'to', 'the', 'international', 'data', 'corporation',
'idc', 'titled', 'the', 'business', 'opportunity', 'of', 'ai',
'reveals', 'a', 'crucial', 'shift', 'businesses', 'are', 'no',
'longer', 'asking', 'if', 'they', 'should', 'integrate', 'ai', 'but',
'rather', 'how', 'to', 'do', 'so', 'strategically', 'to', 'gain', 'a',
'competitive', 'edge']
No Stopwords: ['artificial', 'intelligence', 'ai', 'evolves',
'technological', 'buzzword', 'business', 'imperative', 'recent',
'study', 'commissioned', 'microsoft', 'international', 'data',
'corporation', 'idc', 'titled', 'business', 'opportunity', 'ai',
'reveals', 'crucial', 'shift', 'businesses', 'longer', 'asking',
'integrate', 'ai', 'rather', 'strategically', 'gain', 'competitive',
'edge']
Stemmed: ['artifici', 'intellig', 'ai', 'evolv', 'technolog',
'buzzword', 'busi', 'imper', 'recent', 'studi', 'commiss',
'microsoft', 'intern', 'data', 'corpor', 'idc', 'titl', 'busi',
'opportun', 'ai', 'reveal', 'crucial', 'shift', 'busi', 'longer',
'ask', 'integr', 'ai', 'rather', 'strateg', 'gain', 'competit', 'edg']
Lemmatized: ['artifici', 'intellig', 'ai', 'evolv', 'technolog',
'buzzword', 'busi', 'imper', 'recent', 'studi', 'commiss',
'microsoft', 'intern', 'data', 'corpor', 'idc', 'titl', 'busi',
'opportun', 'ai', 'reveal', 'crucial', 'shift', 'busi', 'longer',
'ask', 'integr', 'ai', 'rather', 'strateg', 'gain', 'competit', 'edg']
```

```
Final_Text: artifici intellig ai evolv technolog buzzword busi imper
recent studi commiss microsoft intern data corpor idc titl busi
opportun ai reveal crucial shift busi longer ask integr ai rather
strateg gain competit edg

==================================================

Sample 3:
Original: A group of Ride for Ceylon participants from the UK and
Canada recently touched down in Colombo, having opted to fly SriLankan
Airlines for the event, which commenced on February 12 and ends today.
Lowercase: a group of ride for ceylon participants from the uk and
canada recently touched down in colombo, having opted to fly srilankan
airlines for the event, which commenced on february 12 and ends today.
No Special Characters: a group of ride for ceylon participants from
the uk and canada recently touched down in colombo  having opted to
fly srilankan airlines for the event  which commenced on february
and ends today
Tokens: ['a', 'group', 'of', 'ride', 'for', 'ceylon', 'participants',
'from', 'the', 'uk', 'and', 'canada', 'recently', 'touched', 'down',
'in', 'colombo', 'having', 'opted', 'to', 'fly', 'srilankan',
'airlines', 'for', 'the', 'event', 'which', 'commenced', 'on',
'february', 'and', 'ends', 'today']
No Stopwords: ['group', 'ride', 'ceylon', 'participants', 'uk',
'canada', 'recently', 'touched', 'colombo', 'opted', 'fly',
'srilankan', 'airlines', 'event', 'commenced', 'february', 'ends',
'today']
Stemmed: ['group', 'ride', 'ceylon', 'particip', 'uk', 'canada',
'recent', 'touch', 'colombo', 'opt', 'fli', 'srilankan', 'airlin',
'event', 'commenc', 'februari', 'end', 'today']
Lemmatized: ['group', 'ride', 'ceylon', 'particip', 'uk', 'canada',
'recent', 'touch', 'colombo', 'opt', 'fli', 'srilankan', 'airlin',
'event', 'commenc', 'februari', 'end', 'today']
Final_Text: group ride ceylon particip uk canada recent touch colombo
opt fli srilankan airlin event commenc februari end today

==================================================

Sample 4:
Original: The ASPI closed in green as a result of price gains in
counters such as Ceylinco Holdings, Melstacorp and Hayleys with the
turnover crossing Rs. 4.8 Bn. A similar behaviour was witnessed in the
S&P SL20. High net worth and institutional investor participation was
noted in Lion Brewery, Ambeon Capital and Ceylinco Holdings. Mixed
interest was observed in Hatton National Bank, R I L Property and
Hemas Holdings whilst retail interest was noted in He
Lowercase: the aspi closed in green as a result of price gains in
counters such as ceylinco holdings, melstacorp and hayleys with the
turnover crossing rs. 4.8 bn. a similar behaviour was witnessed in the
s&p sl20. high net worth and institutional investor participation was
```

noted in lion brewery, ambeon capital and ceylinco holdings. mixed interest was observed in hatton national bank, r i l property and hemas holdings whilst retail interest was noted in he
No Special Characters: the aspi closed in green as a result of price gains in counters such as ceylinco holdings melstacorp and hayleys with the turnover crossing rs   bn a similar behaviour was witnessed in the s p sl   high net worth and institutional investor participation was noted in lion brewery ambeon capital and ceylinco holdings mixed interest was observed in hatton national bank r i l property and hemas holdings whilst retail interest was noted in he
Tokens: ['the', 'aspi', 'closed', 'in', 'green', 'as', 'a', 'result', 'of', 'price', 'gains', 'in', 'counters', 'such', 'as', 'ceylinco', 'holdings', 'melstacorp', 'and', 'hayleys', 'with', 'the', 'turnover', 'crossing', 'rs', 'bn', 'a', 'similar', 'behaviour', 'was', 'witnessed', 'in', 'the', 's', 'p', 'sl', 'high', 'net', 'worth', 'and', 'institutional', 'investor', 'participation', 'was', 'noted', 'in', 'lion', 'brewery', 'ambeon', 'capital', 'and', 'ceylinco', 'holdings', 'mixed', 'interest', 'was', 'observed', 'in', 'hatton', 'national', 'bank', 'r', 'i', 'l', 'property', 'and', 'hemas', 'holdings', 'whilst', 'retail', 'interest', 'was', 'noted', 'in', 'he']
No Stopwords: ['aspi', 'closed', 'green', 'result', 'price', 'gains', 'counters', 'ceylinco', 'holdings', 'melstacorp', 'hayleys', 'turnover', 'crossing', 'rs', 'bn', 'similar', 'behaviour', 'witnessed', 'p', 'sl', 'high', 'net', 'worth', 'institutional', 'investor', 'participation', 'noted', 'lion', 'brewery', 'ambeon', 'capital', 'ceylinco', 'holdings', 'mixed', 'interest', 'observed', 'hatton', 'national', 'bank', 'r', 'l', 'property', 'hemas', 'holdings', 'whilst', 'retail', 'interest', 'noted']
Stemmed: ['aspi', 'close', 'green', 'result', 'price', 'gain', 'counter', 'ceylinco', 'hold', 'melstacorp', 'hayley', 'turnov', 'cross', 'rs', 'bn', 'similar', 'behaviour', 'wit', 'p', 'sl', 'high', 'net', 'worth', 'institut', 'investor', 'particip', 'note', 'lion', 'breweri', 'ambeon', 'capit', 'ceylinco', 'hold', 'mix', 'interest', 'observ', 'hatton', 'nation', 'bank', 'r', 'l', 'properti', 'hema', 'hold', 'whilst', 'retail', 'interest', 'note']
Lemmatized: ['aspi', 'close', 'green', 'result', 'price', 'gain', 'counter', 'ceylinco', 'hold', 'melstacorp', 'hayley', 'turnov', 'cross', 'r', 'bn', 'similar', 'behaviour', 'wit', 'p', 'sl', 'high', 'net', 'worth', 'institut', 'investor', 'particip', 'note', 'lion', 'breweri', 'ambeon', 'capit', 'ceylinco', 'hold', 'mix', 'interest', 'observ', 'hatton', 'nation', 'bank', 'r', 'l', 'properti', 'hema', 'hold', 'whilst', 'retail', 'interest', 'note']
Final_Text: aspi close green result price gain counter ceylinco hold melstacorp hayley turnov cross r bn similar behaviour wit p sl high net worth institut investor particip note lion breweri ambeon capit ceylinco hold mix interest observ hatton nation bank r l properti hema hold whilst retail interest note

```
==================================================


# Apply the function to the entire dataset
df['preprocessed_content'] = df['content'].apply(lambda x:
preprocess_text(x)['Final_Text'])

# Save the preprocessed dataset
preprocessed_file_path =
"/content/drive/MyDrive/Preprocessed_Daily_Mirror_News.xlsx"
df.to_excel(preprocessed_file_path, index=False)

print("Preprocessed dataset saved at:", preprocessed_file_path)
```

```
Preprocessed dataset saved at:
/content/drive/MyDrive/Preprocessed_Daily_Mirror_News.xlsx
```

```
#checking the output
df.head(10)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 1015,\n  \"fields\":
[\n    {\n      \"column\": \"Unnamed: 0\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 59,\n        \"min\": 0,\n
\"max\": 209,\n        \"num_unique_values\": 210,\n
\"samples\": [\n          30,\n          174,\n          85\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"content\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 1015,\n        \"samples\": [\n
\"The Sri Lanka Netball Team will leave for India today to participate
in the 13th Asian Senior Netball Championship 2024, which will be held
from October 18 to 27 at the Koramangala Indoor Stadium in
Bengaluru.\",\n          \"An informal decision is said to have been
taken to keep away a certain prominent Diyawanna member from all
government meetings and other collective activity.\",\n          \"The
first part of this article published last week briefly outlined the
history of Sri Lanka\\u2019s national anthem being sung in Tamil. In
this second and final part, the focus would be on the regress and
progress of the national anthem rendition in Tamil over the years. As
stated earlier the Tamil national anthem issue is in a sense
symptomatic of the escalating ethnic crisis in Sri Lanka.\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"class\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 5,\n        \"samples\": [\n
\"Opinion\",\n          \"World_news\",\n
\"Political_gossip\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"word_count\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 14,\n        \"min\": 3,\n
```

```
\"max\": 87,\n        \"num_unique_values\": 77,\n        \"samples\":
[\n          75,\n          19,\n          18\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"preprocessed_content\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 1014,\n        \"samples\": [\n
\"rafael nadal announc thursday retir profession tenni davi cup final
novemb end career grand slam titl olymp singl gold\",\n
\"certain top team member alway riddl polit observ\",\n
\"everi year winter arriv thousand migratori bird fli south search
food water mani find temporari home mannar island sri lanka largest
island\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


Task 3: Select a Hugging Face Model

```
#!pip install evaluate
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# finetuning the model
from transformers import TrainingArguments, Trainer
from transformers import DistilBertTokenizer,
DistilBertForSequenceClassification
from transformers import EarlyStoppingCallback
from datasets import Dataset

# model evaluation
import evaluate

# hugging face login
from huggingface_hub import notebook_login
from transformers import AutoModelForSequenceClassification,
AutoTokenizer
```

```
Collecting evaluate
  Downloading evaluate-0.4.3-py3-none-any.whl.metadata (9.2 kB)
Collecting datasets>=2.0.0 (from evaluate)
  Downloading datasets-3.5.0-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (2.0.2)
Collecting dill (from evaluate)
  Downloading dill-0.3.9-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in
/usr/local/lib/python3.11/dist-packages (from evaluate) (2.2.2)
Requirement already satisfied: requests>=2.19.0 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (2.32.3)
```

```
Requirement already satisfied: tqdm>=4.62.1 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (4.67.1)
Collecting xxhash (from evaluate)
  Downloading xxhash-3.5.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess (from evaluate)
  Downloading multiprocess-0.70.17-py311-none-any.whl.metadata (7.2
kB)
Requirement already satisfied: fsspec>=2021.05.0 in
/usr/local/lib/python3.11/dist-packages (from fsspec[http]>=2021.05.0-
>evaluate) (2025.3.0)
Requirement already satisfied: huggingface-hub>=0.7.0 in
/usr/local/lib/python3.11/dist-packages (from evaluate) (0.29.3)
Requirement already satisfied: packaging in
/usr/local/lib/python3.11/dist-packages (from evaluate) (24.2)
Requirement already satisfied: filelock in
/usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0-
>evaluate) (3.18.0)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0-
>evaluate) (18.1.0)
Collecting dill (from evaluate)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Collecting multiprocess (from evaluate)
  Downloading multiprocess-0.70.16-py311-none-any.whl.metadata (7.2
kB)
Collecting fsspec>=2021.05.0 (from fsspec[http]>=2021.05.0->evaluate)
  Downloading fsspec-2024.12.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: aiohttp in
/usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0-
>evaluate) (3.11.14)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0-
>evaluate) (6.0.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.7.0-
>evaluate) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.19.0-
>evaluate) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.19.0-
>evaluate) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.19.0-
>evaluate) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.19.0-
>evaluate) (2025.1.31)
```

```
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas->evaluate)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.11/dist-packages (from pandas->evaluate)
(2025.1)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.11/dist-packages (from pandas->evaluate)
(2025.1)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (6.2.0)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (0.3.0)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp-
>datasets>=2.0.0->evaluate) (1.18.3)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2-
>pandas->evaluate) (1.17.0)
Downloading evaluate-0.4.3-py3-none-any.whl (84 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 84.0/84.0 kB 3.8 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 491.2/491.2 kB 14.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 9.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 183.9/183.9 kB 14.0 MB/s eta
0:00:00
ultiprocess-0.70.16-py311-none-any.whl (143 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 11.0 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 14.4 MB/s eta
0:00:00
ultiprocess, datasets, evaluate
```

```
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2025.3.0
    Uninstalling fsspec-2025.3.0:
      Successfully uninstalled fsspec-2025.3.0
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
gcsfs 2025.3.0 requires fsspec==2025.3.0, but you have fsspec
2024.12.0 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cublas-cu12 12.5.3.2 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-cupti-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-nvrtc-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-runtime-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-runtime-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cudnn-cu12==9.1.0.70;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cudnn-cu12 9.3.0.75 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cufft-cu12 11.2.3.61 which is incompatible.
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-curand-cu12 10.3.6.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cusolver-cu12 11.6.3.83 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cusparse-cu12 12.5.1.3 which is incompatible.
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-nvjitlink-cu12 12.5.82 which is incompatible.
Successfully installed datasets-3.5.0 dill-0.3.8 evaluate-0.4.3
fsspec-2024.12.0 multiprocess-0.70.16 xxhash-3.5.0
```

```
#conversion of categorical values to numerical

# Import LabelEncoder
from sklearn.preprocessing import LabelEncoder

# create a LabelEncoder object
le = LabelEncoder()
```

```
# encode the sentiment column as 0 for positive and 1 for negative
df['class'] = le.fit_transform(df['class'])

# print the unique values of the encoded column
print(df['class'].unique())

print(le.classes_)
```

```
[0 1 2 3 4]
['Business' 'Opinion' 'Political_gossip' 'Sports' 'World_news']
```

Model Selected: distilbert-base-uncased

```
# tokenizer and model
model_name = "distilbert-base-uncased"
tokenizer = DistilBertTokenizer.from_pretrained(model_name)
model =
DistilBertForSequenceClassification.from_pretrained(model_name,
num_labels=5, ignore_mismatched_sizes=True) # Added
ignore_mismatched_sizes=True
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/
_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(
```


```
{"model_id":"e087c2a13402460dba9eba768f7b2be9","version_major":2,"vers
ion_minor":0}

{"model_id":"32728c802eeb41c9b1d66d1102cf1c13","version_major":2,"vers
ion_minor":0}

{"model_id":"780a2835c6604a65a4cbde0d14ed5315","version_major":2,"vers
ion_minor":0}

{"model_id":"837435630f634291b60bd064fb0f42a9","version_major":2,"vers
ion_minor":0}

{"model_id":"abb10aba4aa34b17880a46462e73afb3","version_major":2,"vers
ion_minor":0}
```


```
Some weights of DistilBertForSequenceClassification were not
initialized from the model checkpoint at distilbert-base-uncased and
```

```
are newly initialized: ['classifier.bias', 'classifier.weight',
'pre_classifier.bias', 'pre_classifier.weight']
You should probably TRAIN this model on a down-stream task to be able
to use it for predictions and inference.
```

Justification: Selection of distilbert-base-uncased for News Classification & Question Answering

---

We selected distilbert-base-uncased for news classification and question answering due to its efficiency, contextual understanding, and strong performance while being lighter and faster than the original BERT model.

Why DistilBERT?

1. Proven Performance – DistilBERT retains 97% of BERT's accuracy while being 60% faster and 40% smaller, making it well-suited for real-time applications like ours.
2. Contextual Understanding – Processes text bidirectionally, ensuring better comprehension of news articles and accurate answer extraction.
3. Pre-trained on Large Datasets – Trained on BooksCorpus and Wikipedia, providing a strong foundation in general language understanding, improving classification and QA accuracy.
4. Easily Fine-tunable – Supports multi-class classification (Business, Opinion, Political Gossip, Sports, World News) and can be directly used for question answering without additional training.
5. Multiclass Classification Support – Uses a feed-forward layer for multi-class classification, making it ideal for categorizing diverse news topics.
6. Question Answering Capability – Fine-tuned on SQuAD (Stanford Question Answering Dataset), making it effective in extracting relevant answers from news content.
7. Uncased Model Advantage – Being case-insensitive, it treats words equally, improving generalization and simplifying preprocessing.
8. Efficiency & Speed – Smaller and faster than BERT, allowing real-time inference, making it perfect for web apps and cloud deployment.
9. Ease of Integration into Web Apps –

   Hugging Face Transformers Library: Easily integrates into web apps for real-time classification & QA.

   Scalability: Can be deployed on local servers or cloud environments efficiently.

Task 4: Finetune a Pre-trained Hugging Face Model

```
# train and validation split

X_train, X_val, y_train, y_val =
train_test_split(df["preprocessed_content"], df["class"],

test_size=0.2,random_state=42,

stratify=df["class"])

# Converting the original df to a data structure
train_dataset = Dataset.from_dict({"preprocessed_content":
list(X_train), "label": list(y_train)})
val_dataset = Dataset.from_dict({"preprocessed_content": list(X_val),
"label": list(y_val)})

# Tokenize the dataset
'''
> Here, we are using the transformer architecture.
> We give unique IDs to each token.
> Introduce special tokens.
> Padding to convert sentences to equal length.
> Finally, introduce an attention mask
'''
def tokenize_function(review):
    return tokenizer(review["preprocessed_content"],
padding="max_length", truncation=True, max_length=512)

tokenized_train_dataset = train_dataset.map(tokenize_function,
batched=True)
tokenized_val_dataset = val_dataset.map(tokenize_function,
batched=True)

# Access the correct keys in the tokenized dataset
print(tokenized_train_dataset[5]['preprocessed_content'] , '\n')
print(tokenized_train_dataset[0]['input_ids'], '\n')
print(tokenized_train_dataset[0]['attention_mask'], '\n')
```


```
{"model_id":"25b99b6a237e432b92631a25940c2d2b","version_major":2,"vers
ion_minor":0}

{"model_id":"71092b37a545493fa9cf25ced2b21006","version_major":2,"vers
ion_minor":0}
```


```
sri lanka west indi clash second odi three match seri tomorrow
pallekel intern cricket stadium kandi

[101, 4957, 14085, 3126, 4942, 5332, 9032, 3089, 25022, 2278, 2907,
```

```
3404, 2171, 2569, 27849, 6887, 27292, 10732, 4904, 27849, 2740, 10010,
2711, 2729, 4031, 7117, 1037, 10513, 2099, 7178, 9866, 19976, 2715,
16596, 2368, 2278, 2165, 2112, 14855, 4246, 2532, 25204, 3119, 4189,
10147, 24475, 2218, 3522, 2093, 9530, 3366, 12690, 2154, 14163, 4779,
2527, 2860, 2598, 14855, 4246, 2532, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0]

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0, 0, 0, 0, 0, 0]


# Load metric

metric = evaluate.combine(["accuracy", evaluate.load("precision",
average="weighted"), evaluate.load("recall", average="weighted"),
evaluate.load("f1", average="weighted")])

def compute_metrics(p):
  '''
  This function will calculate the values for the above-mentioned
metrics
  by comparing predicted values with real values(references).
  '''
  return metric.compute(predictions=np.argmax(p.predictions, axis=1),
references=p.label_ids)

{"model_id":"485e18cd1f584ad2b6d60413ed655483","version_major":2,"vers
ion_minor":0}

{"model_id":"3bd9217d0fe14b9c9884514727473991","version_major":2,"vers
ion_minor":0}

{"model_id":"6f4c0c698440495b98c7002de73a9ed1","version_major":2,"vers
ion_minor":0}

{"model_id":"73454eca1c9e458688ebd6524be698f2","version_major":2,"vers
ion_minor":0}

# Trainer setup
args = TrainingArguments(
    output_dir="HuggingFaceAttempt1",
    run_name="version1",
    evaluation_strategy="steps",

    eval_steps=500,
    per_device_train_batch_size= 4,
    per_device_eval_batch_size=4,
    num_train_epochs=1,
    seed=0,
    load_best_model_at_end=True,
)

trainer = Trainer(
    model=model,
    args=args,
    train_dataset=tokenized_train_dataset,
    eval_dataset=tokenized_val_dataset,
    compute_metrics=compute_metrics,
    callbacks=[EarlyStoppingCallback(early_stopping_patience=3)],
```

```
)
model.config.id2label = {
    0: "Business",
    1: "Opinion",
    2: "Political_gossip",
    3: "Sports",
    4: "World_news"
}
model.config.label2id = {v: k for k, v in
model.config.id2label.items()}
```

```
/usr/local/lib/python3.11/dist-packages/transformers/
training_args.py:1611: FutureWarning: `evaluation_strategy` is
deprecated and will be removed in version 4.46 of  Transformers. Use
`eval_strategy` instead
  warnings.warn(
```

```
# to get rid of Weights and Bias login
# W&B will help you keep track of a model training
import os
os.environ["WANDB_DISABLED"] = "true"
import wandb

# Train pre-trained model
train_output = trainer.train()
# Save trained model
trainer.save_model("content/")
# Print the train output
print(train_output)
```

```
wandb: Using wandb-core as the SDK backend.  Please refer to
https://wandb.me/wandb-core for more information.

<IPython.core.display.Javascript object>

wandb: Logging into wandb.ai. (Learn how to deploy a W&B server
locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here:
https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter:wandb: WARNING
If you're specifying your api key in code, ensure this code is not
shared publicly.
wandb: WARNING Consider setting the WANDB_API_KEY environment
variable, or running `wandb login` from the command line.
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Currently logged in as: venuja-udukumbura (venuja-udukumbura-
university-of-moratuwa) to https://api.wandb.ai. Use `wandb login --
relogin` to force relogin

<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

TrainOutput(global_step=203, training_loss=0.9099612682323738,
metrics={'train_runtime': 3154.9778, 'train_samples_per_second':
0.257, 'train_steps_per_second': 0.064, 'total_flos':
107569282437120.0, 'train_loss': 0.9099612682323738, 'epoch': 1.0})

# Login to Hugging Face
notebook_login()

# Save model and tokenizer locally
model.save_pretrained("News_class_classification")
tokenizer.save_pretrained("News_class_classification")

{"model_id":"7e80dd51fe544ce584ee12c232586132","version_major":2,"vers
ion_minor":0}

('News_class_classification/tokenizer_config.json',
 'News_class_classification/special_tokens_map.json',
 'News_class_classification/vocab.txt',
 'News_class_classification/added_tokens.json')

# Push to Hugging Face Hub
model.push_to_hub("Venuja-Udukumbura/News_class_classification")
tokenizer.push_to_hub("Venuja-Udukumbura/News_class_classification")

{"model_id":"ac1e4c676fe24a8b9c7cb288f1282947","version_major":2,"vers
ion_minor":0}

{"model_id":"86524cb9c6ff4adeb403912fec8b2eba","version_major":2,"vers
ion_minor":0}

No files have been modified since last commit. Skipping to prevent
empty commit.
WARNING:huggingface_hub.hf_api:No files have been modified since last
commit. Skipping to prevent empty commit.

{"type":"string"}

# get the model
new_model = "Venuja-Udukumbura/News_class_classification"
```

```
# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("text-classification",
model="Venuja-Udukumbura/News_class_classification")

#testing the model
text="Truth, the idiom goes is often stranger than fiction. Our
country seems to embody this truism. Days ago we were jolted when
media reported the Attorney General -whose independent status has
withstood numerous changes as per the Constitution- advised law
enforcement authorities that there was insufficient evidence to
prosecute the three persons accused of abducting the driver of -
Lasantha Wickrematunge- Editor of the now defunct ‘Sunday Leader’ n"

pipe (text)
```

```
[{'label': 'Opinion', 'score': 0.6141933798789978}]
```

LINKS

Dear Miss,

You might not be able to access our model from the provided link because our fine-tuned Hugging Face model is private. We initially attempted to make it public, but all of us in the group encountered errors. The solution we found was to keep the model private and use a token in our code to integrate it with the web app. However, the app can be viewed without any issue.

Apologies for any inconvenience this may cause. If there's any issue, please let us know, and we can show the model in person if needed.

Link to the fine-tuned model

https://huggingface.co/Venuja-Udukumbura/News_class_classification

Link to the WebApp

https://huggingface.co/spaces/Venuja-Udukumbura/News-Classification-App