

```
# 1 Install required libraries
!pip install nltk xgboost

# 2 Import libraries
import pandas as pd
import numpy as np
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

# Download NLTK resources
nltk.download("stopwords")
nltk.download("punkt_tab")
nltk.download("wordnet")

# 3 Load datasets (upload Fake.csv and True.csv manually)
from google.colab import files
uploaded = files.upload()
fake = pd.read_csv("Fake.csv", on_bad_lines='skip',
real = pd.read_csv("True.csv", on_bad_lines='skip',

print("✓ Fake dataset shape:", fake.shape)
print("✓ Real dataset shape:", real.shape)

# 4 Add labels
fake["label"] = 1    # Fake news
real["label"] = 0    # Real news

# Combine datasets
df = pd.concat([fake[["text", "label"]], real[["text", "label"]]])
print("✓ Combined dataset shape:", df.shape)

# 5 Preprocessing
stop_words = set(stopwords.words("english"))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    if not isinstance(text, str):
        return ""
    text = text.lower()
    text = re.sub(r"[^a-z\s]", "", text)
    tokens = nltk.word_tokenize(text)
    tokens = [lemmatizer.lemmatize(word) for word in tokens]
    return " ".join(tokens)

df["clean_text"] = df["text"].apply(preprocess_text)
print("✓ Preprocessing done")
print(df[["text", "clean_text"]].head(3))
```

```
# 6 Train-test split and vectorization
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

X = df["clean_text"]
y = df["label"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

vectorizer = TfidfVectorizer(max_df=0.7)
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

print("✅ Vectorization complete:", X_train_vec.shape)

# 7 XGBoost Classifier
from xgboost import XGBClassifier
from sklearn.metrics import classification_report, accuracy_score

xgb_model = XGBClassifier(use_label_encoder=False, n_estimators=100)
xgb_model.fit(X_train_vec, y_train)

y_pred = xgb_model.predict(X_test_vec)

# 8 Evaluation
accuracy = accuracy_score(y_test, y_pred)
print("✅ XGBoost Accuracy:", accuracy)
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: xgboost in /usr/local/lib/python3.12/dist-
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-p
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: nvidia-nccl-cu12 in /usr/local/lib/python3
Requirement already satisfied: scipy in /usr/local/lib/python3.12/dist-pa
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving True.csv to True.csv

Saving Fake.csv to Fake.csv

- Fake dataset shape: (23481, 4)
- Real dataset shape: (21417, 4)
- Combined dataset shape: (44898, 2)
- Preprocessing done

```
text \
0 Donald Trump just couldn't wish all Americans ...
1 House Intelligence Committee Chairman Devin Nu...
2 On Friday, it was revealed that former Milwauk...
```

```
clean_text
0 donald trump wish american happy new year leav...
1 house intelligence committee chairman devin nu...
2 friday revealed former milwaukee sheriff david...
```

Vectorization complete: (35918, 181281)
/usr/local/lib/python3.12/dist-packages/xgboost/training.py:183: UserWarn
Parameters: { "use_label_encoder" } are not used.

```
bst.update(dtrain, iteration=i, fobj=obj)
 XGBoost Accuracy: 0.9978841870824053
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4247
1	1.00	1.00	1.00	4733
accuracy			1.00	8980
macro avg	1.00	1.00	1.00	8980
weighted avg	1.00	1.00	1.00	8980