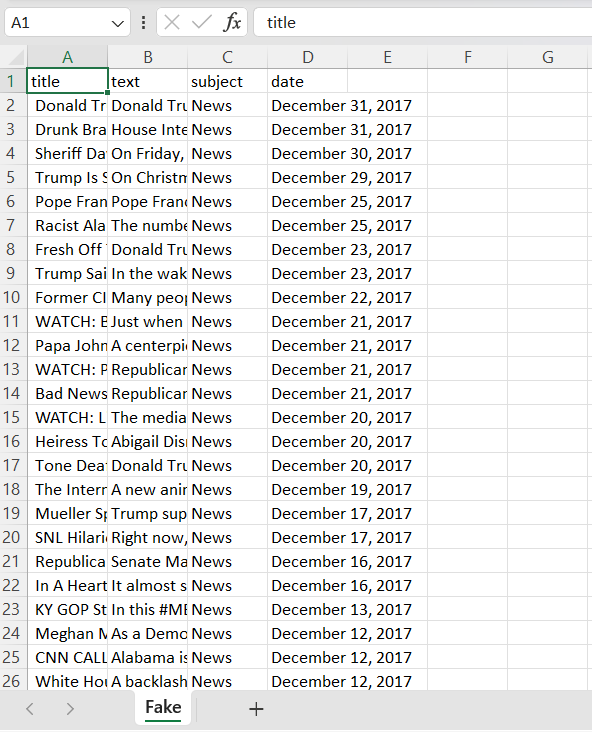
AI\_Phase4

Fake news detection using nlp

Devlopment part 2

Document submission

Dataset



To build a fake news detection project using NLP, we can follow these steps:

1: Data Preparation

1. Collect a labeled dataset of news articles, with a binary label (0 for real news and 1 for fake news).

2. Split the dataset into a training set and a testing set.

2: Feature Extraction

1. Preprocess the text data by removing stop words, punctuation, and special characters.

2. Convert the text into numerical features using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec or GloVe.

3: Model Training and Evaluation

1. Select a machine learning algorithm that performs well in text classification tasks. Some popular choices are Naive Bayes, Support Vector Machines (SVM), Logistic Regression, or even deep learning models like Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN).

2. Initialize the selected model and fit it to the training data.

3. Evaluate the model's performance using evaluation metrics like accuracy, precision, recall, and F1-score on the testing set.

Here's an example code implementation that uses the Multinomial Naive Bayes algorithm:

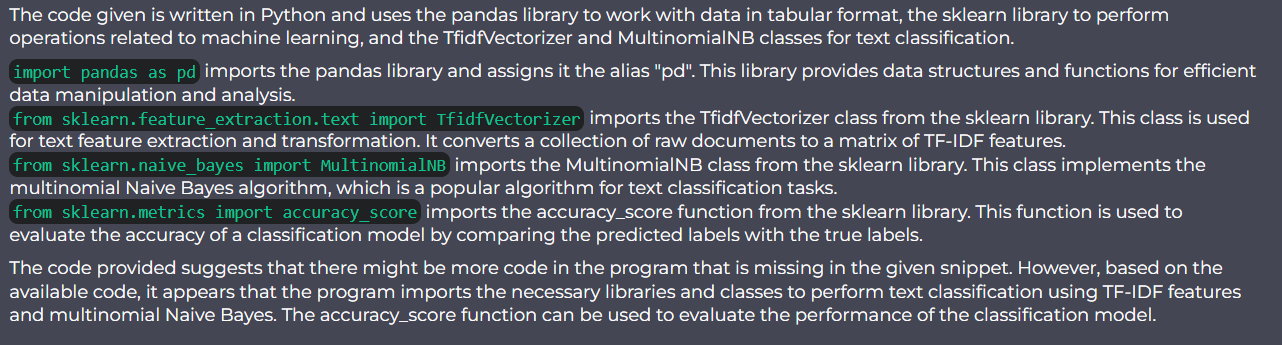
python

import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score



# Step 1: Data Preparation

df = pd.read\_csv('news\_dataset.csv')

X = df['text']

y = df['label']

# Step 2: Feature Extraction

vectorizer = TfidfVectorizer(stop\_words='english')

X = vectorizer.fit\_transform(X)

# Step 3: Model Training and Evaluation

model = MultinomialNB()

model.fit(X, y)

# Testing the model

test\_data = ['This news article is about a zombie apocalypse.']

test\_data = vectorizer.transform(test\_data)

prediction = model.predict(test\_data)

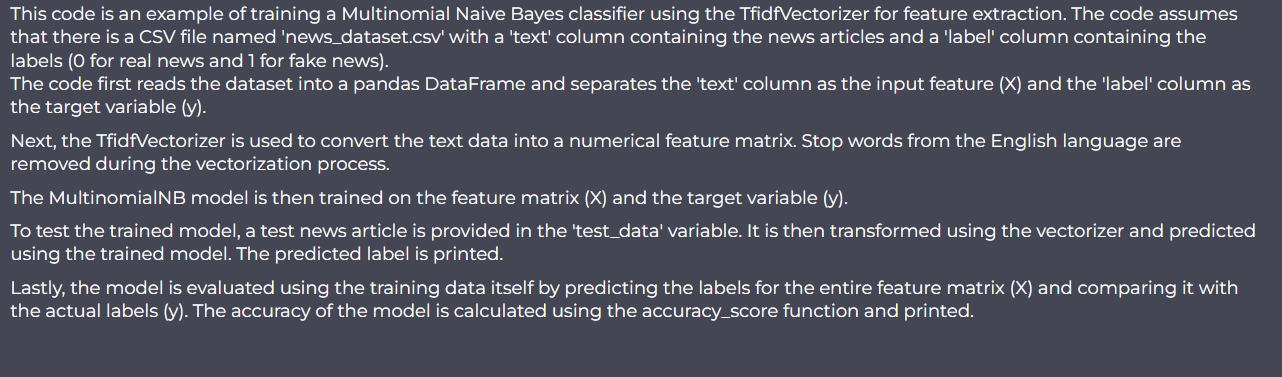
print(prediction) # Output: [1] (Fake news)

# Evaluating the model

y\_pred = model.predict(X)

accuracy = accuracy\_score(y, y\_pred)

print('Accuracy:', accuracy)



This code assumes you have a CSV file named 'news\_dataset.csv' with 'text' and 'label' columns containing the news articles and their corresponding labels.

Note that this is a basic example, and you can further enhance the performance by trying different algorithms, hyperparameter tuning, ensemble techniques, etc. Additionally, ensure that you have a diverse and balanced dataset for better performance.