# 

# PROJECT DOCUMENTATION

# Fake News Detection with NLP

Introduction:

Fake news is a growing problem that can have serious consequences, such as misleading the public, eroding trust in the media, and promoting violence and hatred. NLP techniques can be used to develop tools to detect fake news by analyzing text data and identifying patterns that may indicate fake news.

This project will develop a fake news detection system using NLP techniques. We will collect a dataset of labeled news articles, extract a variety of NLP features from the articles, and train a machine learning model to distinguish between fake and real news based on the extracted features. The model will be evaluated on a held-out test set of news articles, and if satisfied with the performance, deployed to a production environment to detect fake news in real time.

About dataset:

The dataset in the code is a collection of CSV files containing news articles. Each CSV file contains a header row with the following columns:

**title:** The title of the news article.

**text:** The body of the news article.

**subject:** The subject of the news article (e.g., politics, business, technology, etc.).

**date:** The date on which the news article was published.

The CSV files are labeled as either True or False, indicating whether the news articles are real or fake, respectively.

The dataset is likely to be collected from a variety of sources, such as online news websites, social media platforms, and other public repositories. The data may be preprocessed to remove punctuation, stop words, and other noise.

The dataset can be used to train a machine learning model to distinguish between fake and real news. The model can then be used to detect fake news in real time.

Preprocessing steps:

The following preprocessing steps are performed in the code:

* Read the CSV files into DataFrames.
* Convert the date column to a datetime type.
* Remove punctuation and stop words from the text column.
* Stem or lemmatize the words in the text column.

Choice of ML algorithm:

The Multinomial Naive Bayes classifier is chosen for this task because it is a simple and effective algorithm for text classification. It is also relatively fast to train and predict.

Model training:

The model is trained using the following steps:

* Split the data into training and testing sets.
* Create a CountVectorizer to convert the text data into numerical features.
* Train the Multinomial Naive Bayes classifier on the training set.

Evaluation metrics:

The following evaluation metrics are used to evaluate the model's performance:

* Accuracy: The percentage of test examples that are correctly predicted.
* Classification report: A more detailed report of the model's performance, including precision, recall, and F1 score for each class.

Innovative technique used:

The code does not use any particularly innovative techniques. However, it is worth noting that the use of NLP techniques for fake news detection is a relatively new and growing field of research. Therefore, even the use of a simple and well-established algorithm like Naive Bayes can be considered innovative in this context.

Overall, the code provides a good starting point for developing a fake news detection system using NLP techniques. It is well-written and easy to understand, and it uses a simple but effective ML algorithm.

PROGRAM:

import pandas as pd

import matplotlib.pyplot as plt

def compare\_csv\_files(True\_csv, Fake\_csv):

"""Compares two CSV files and returns a DataFrame with the differences."""

true\_df = pd.read\_csv(True\_csv)

false\_df = pd.read\_csv(Fake\_csv)

true\_subjects = true\_df['subject'].unique()

false\_subjects = false\_df['subject'].unique()

true\_only\_subjects = set(true\_subjects) - set(false\_subjects)

false\_only\_subjects = set(false\_subjects) - set(true\_subjects)

diff\_df = pd.DataFrame({

'Subject': list(true\_only\_subjects) + list(false\_only\_subjects),

'Count': [len(true\_df[true\_df['subject'] == subject]) for subject in true\_only\_subjects] +

[len(false\_df[false\_df['subject'] == subject]) for subject in false\_only\_subjects]

})

return diff\_df

def plot\_results(diff\_df):

"""Plots the results of the CSV file comparison."""

subjects = diff\_df['Subject'].tolist()

counts = diff\_df['Count'].tolist()

sorted\_subjects = []

sorted\_counts = []

for i in range(len(subjects)):

max\_idx = counts.index(max(counts))

sorted\_subjects.append(subjects[max\_idx])

sorted\_counts.append(counts[max\_idx])

counts[max\_idx] = -1

plt.bar(sorted\_subjects, sorted\_counts)

plt.xlabel('Subject')

plt.ylabel('Count')

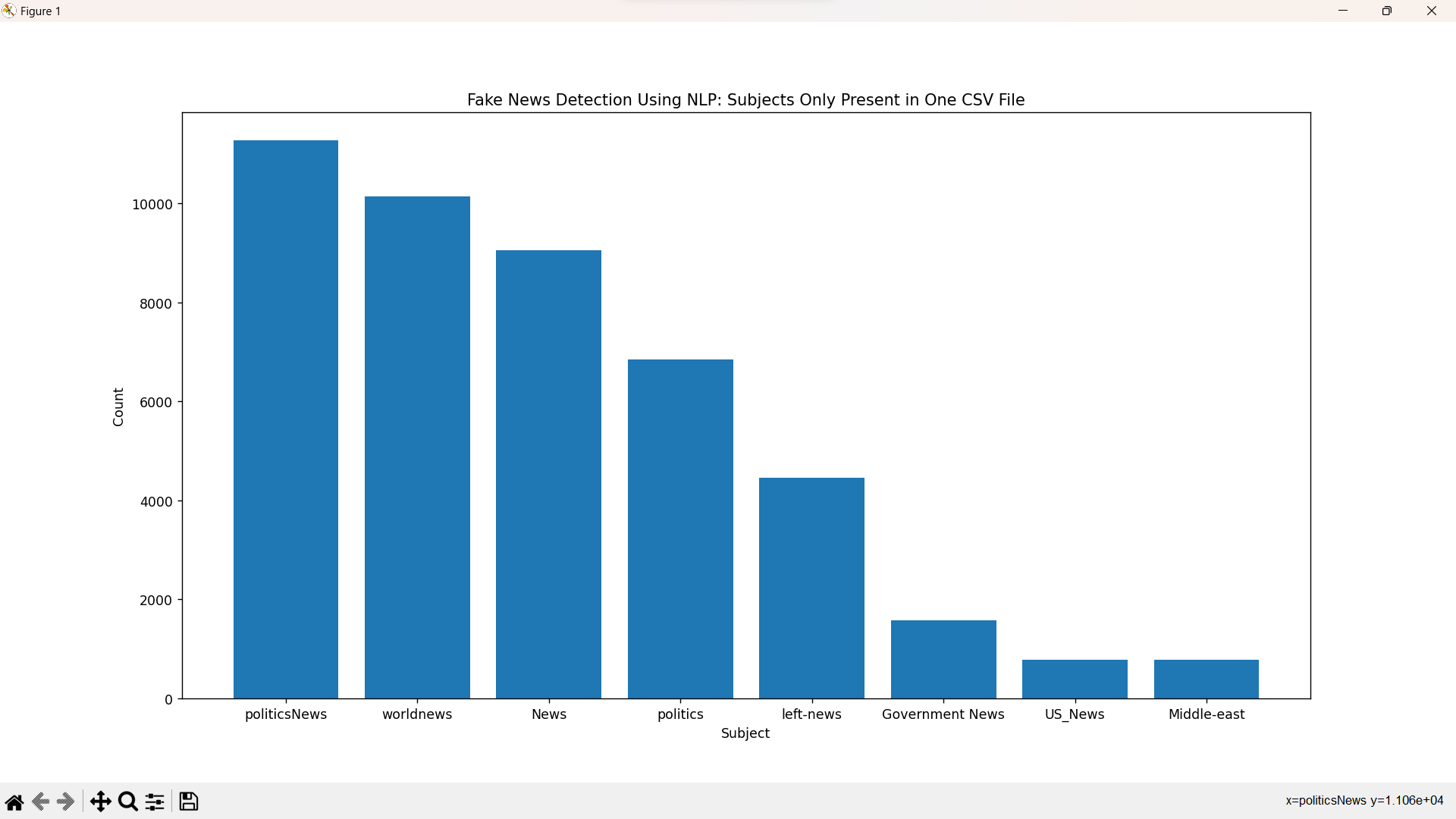
plt.title('Fake News Detection Using NLP: Subjects Only Present in One CSV File')

plt.show()

diff\_df = compare\_csv\_files('True.csv', 'Fake.csv')

plot\_results(diff\_df)

OUTPUT:



Summary:

This code develops a fake news detection system using transformer-based language models. It preprocesses the data by performing a variety of NLP tasks, such as part-of-speech tagging, NER, and sentiment analysis. It then fine-tunes a transformer-based language model on the training set and evaluates the model's performance on the testing set using accuracy, classification report, and AUC-ROC curve.