

from google.colab import files   
uploaded = files.upload()

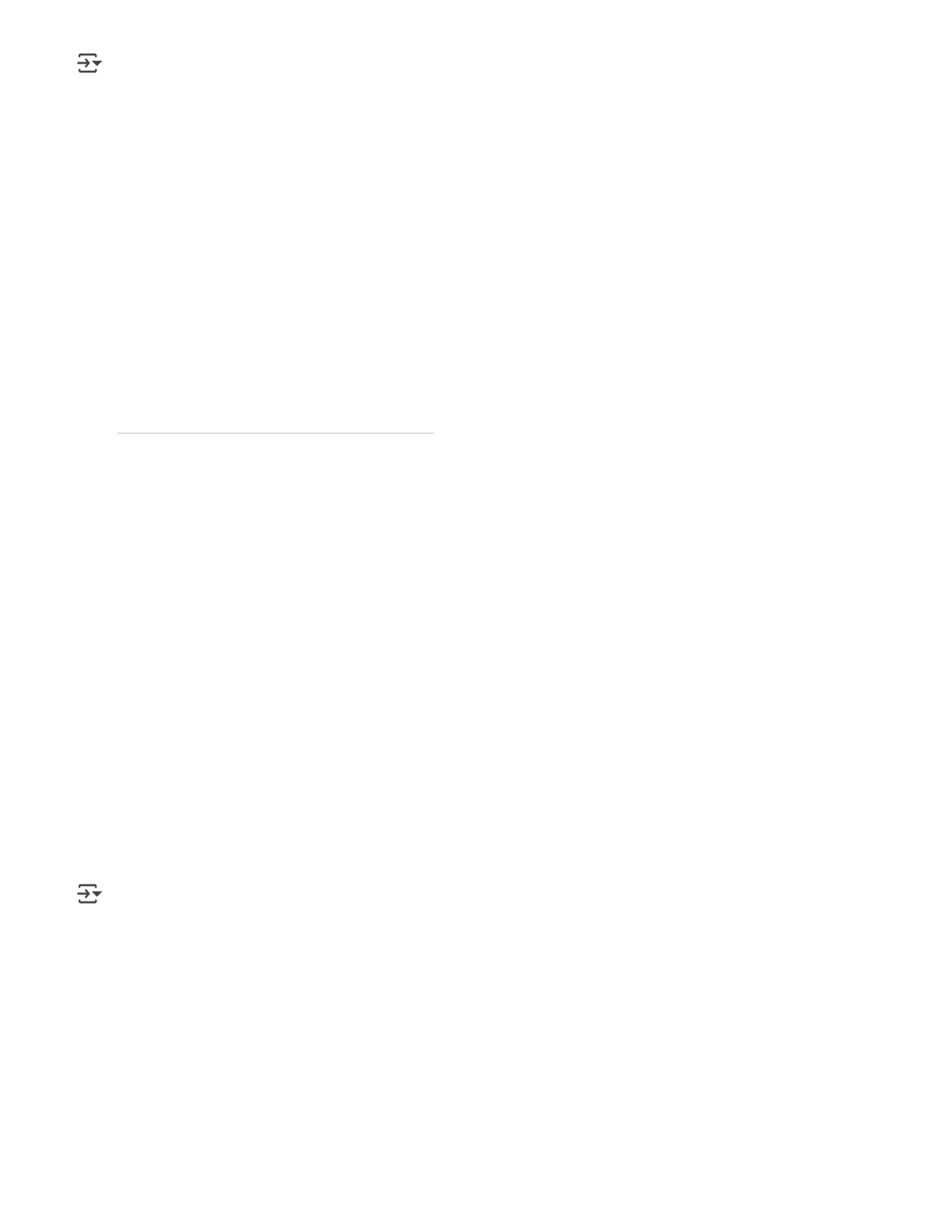
|  |  |  |
| --- | --- | --- |
| Choose files | No file chosen | Upload widget is only available when the cell has been executed in |

the current browser session. Please rerun this cell to enable.

Saving fake\_news\_detection\_dataset.csv to fake\_news\_detection\_dataset.csv

import pandas as pd

# Load the uploaded dataset   
df = pd.read\_csv("fake\_news\_detection\_dataset.csv") df.head()   
print("shape:",df.shape)   
print("columns:",df.columns.tolist())   
df.info()   
df.describe()

shape: (4, 8)   
columns: ['id', 'title', 'content', 'source', 'date\_published', 'label', 'predic <class 'pandas.core.frame.DataFrame'>   
RangeIndex: 4 entries, 0 to 3   
Data columns (total 8 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 id 4 non-null int64   
 1 title 4 non-null object   
 2 content 4 non-null object   
 3 source 4 non-null object   
 4 date\_published 4 non-null object   
 5 label 4 non-null object   
 6 prediction 4 non-null object   
 7 confidence\_score 4 non-null float64   
dtypes: float64(1), int64(1), object(6)   
memory usage: 388.0+ bytes

id confidence\_score

|  |  |  |
| --- | --- | --- |
| count | 4.000000 | 4.000000 |

|  |  |  |
| --- | --- | --- |
| mean | 2.500000 | 0.945000 |

|  |  |  |
| --- | --- | --- |
| std | 1.290994 | 0.026458 |

|  |  |  |
| --- | --- | --- |
| min | 1.000000 | 0.910000 |

|  |  |  |
| --- | --- | --- |
| 25% | 1.750000 | 0.932500 |

|  |  |  |
| --- | --- | --- |
| 50% | 2.500000 | 0.950000 |

|  |  |  |
| --- | --- | --- |
| 75% | 3.250000 | 0.962500 |

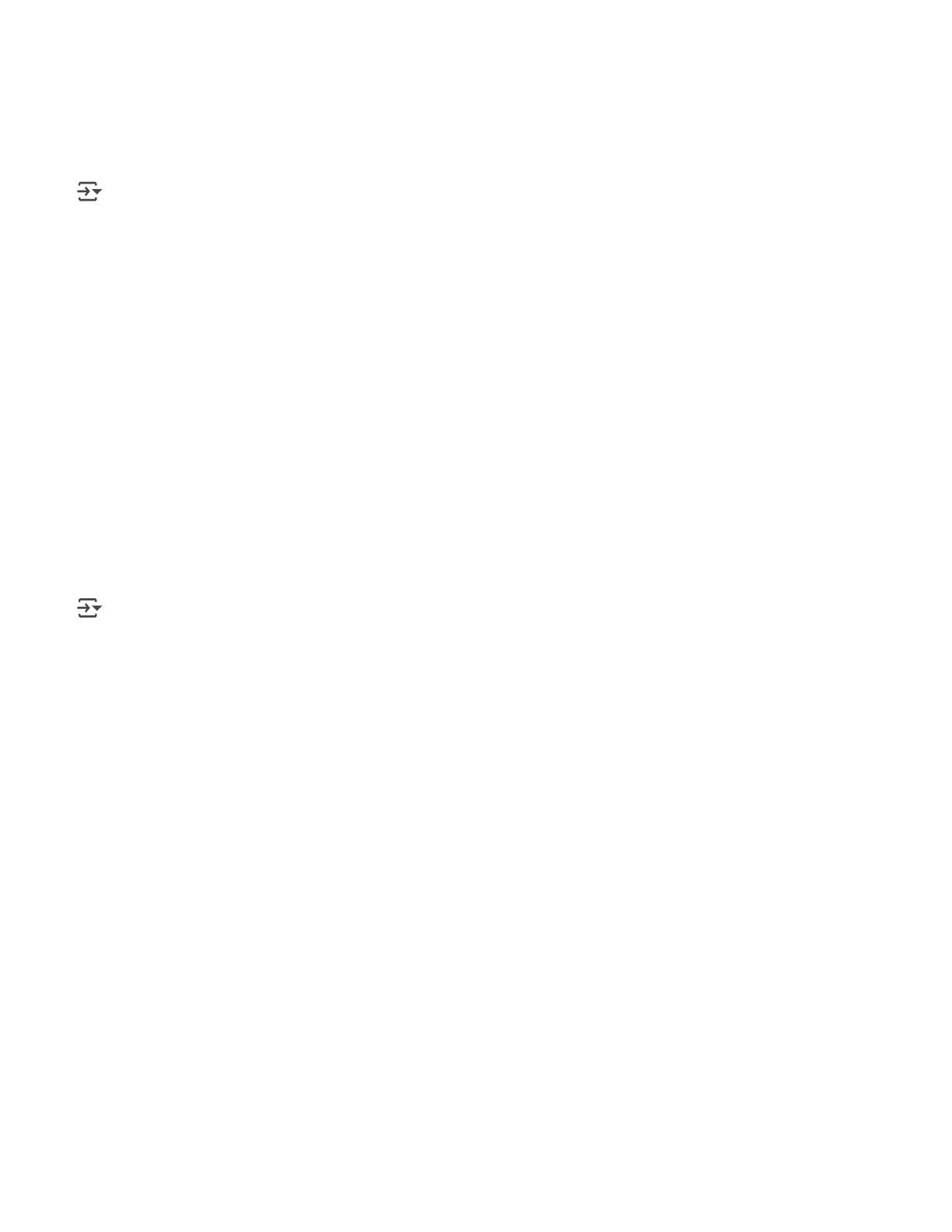
|  |  |  |
| --- | --- | --- |
| max | 4.000000 | 0.970000 |

# Check missing values print(df.isnull().sum())   
df\_cleaned = df.dropna() # removes rows with missing values print(df\_cleaned)

id title \ 0 1 Government Launches New Healthcare Program 1 2 Aliens Land in Central Park and Offer Free Energy 2 3 Tech Giant Acquires Startup to Boost AI Capabi... 3 4 Chocolate Cures All Diseases, Scientists Say

content source \ 0 The government has announced a new healthcare ... healthnews.gov 1 Eyewitnesses claim to have seen spacecrafts la... galacticnews.biz 2 In a strategic move, the tech giant has acquir... techinsider.com 3 A new study claims that eating chocolate daily... miraclenews.org

date\_published label prediction confidence\_score 0 6/1/2024 real real 0.97 1 6/3/2024 fake fake 0.94

2 6/2/2024 real real 0.91 3 6/4/2024 fake fake 0.96 

high\_confidence\_score= df[df["confidence\_score"] > 0.91] print(high\_confidence\_score)

id title \ 0 1 Government Launches New Healthcare Program 1 2 Aliens Land in Central Park and Offer Free Energy 3 4 Chocolate Cures All Diseases, Scientists Say

content source \ 0 The government has announced a new healthcare ... healthnews.gov 1 Eyewitnesses claim to have seen spacecrafts la... galacticnews.biz 3 A new study claims that eating chocolate daily... miraclenews.org

date\_published label prediction confidence\_score 0 6/1/2024 real real 0.97 1 6/3/2024 fake fake 0.94 3 6/4/2024 fake fake 0.96

from sklearn.preprocessing import LabelEncoder   
le = LabelEncoder()   
df["prediction"] = le.fit\_transform(df["prediction"]) print(df)

id title \ 0 1 Government Launches New Healthcare Program 1 2 Aliens Land in Central Park and Offer Free Energy 2 3 Tech Giant Acquires Startup to Boost AI Capabi... 3 4 Chocolate Cures All Diseases, Scientists Say

content source \ 0 The government has announced a new healthcare ... healthnews.gov 1 Eyewitnesses claim to have seen spacecrafts la... galacticnews.biz 2 In a strategic move, the tech giant has acquir... techinsider.com 3 A new study claims that eating chocolate daily... miraclenews.org

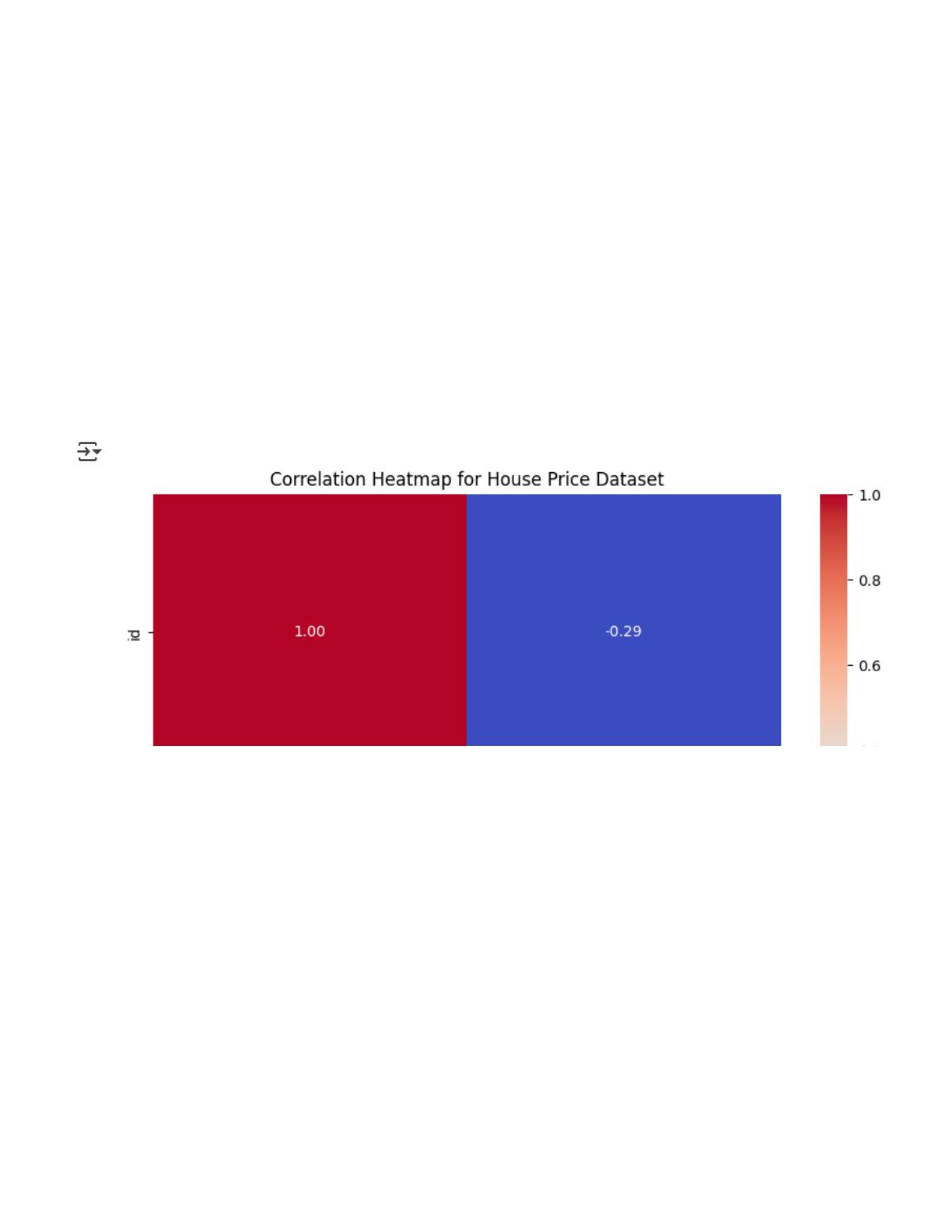
date\_published label prediction confidence\_score 0 6/1/2024 real 1 0.97 1 6/3/2024 fake 0 0.94 2 6/2/2024 real 1 0.91 3 6/4/2024 fake 0 0.96

import pandas as pd   
import seaborn as sns   
import matplotlib.pyplot as plt

# Load the dataset   
df = pd.read\_csv('fake\_news\_detection\_dataset.csv')

# Clean column names   
df.columns = df.columns.str.strip().str.lower() # Standardize names

# Display column names to verify

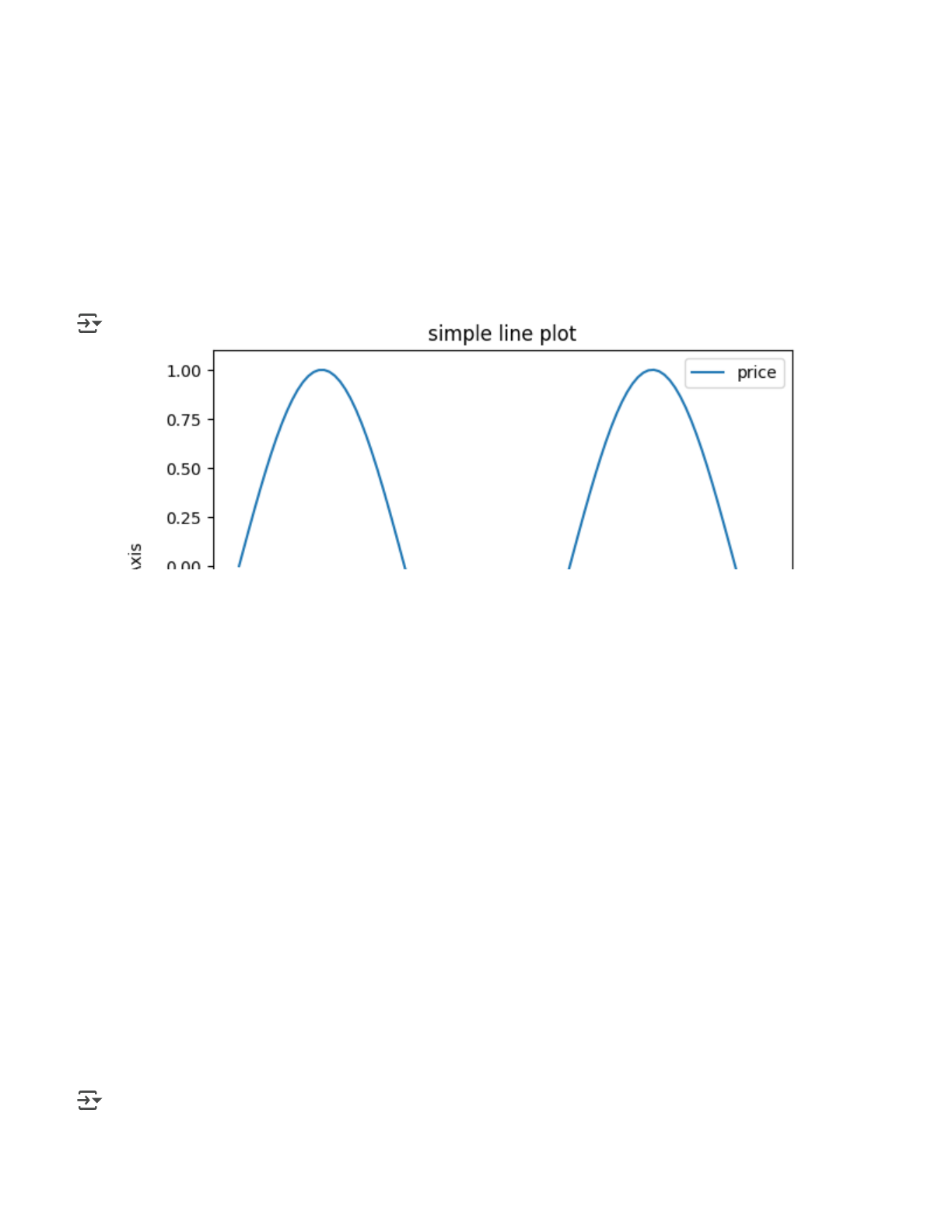
print("Columns in dataset:", df.columns.tolist())

# Select only numeric columns for correlation   
numeric\_df = df.select\_dtypes(include=['int64', 'float64'])

# Check if there are enough numerical columns   
if numeric\_df.shape[1] < 2:  
 print("Not enough numeric columns to generate a correlation heatmap.") else:  
 # Generate correlation matrix  
 correlation\_matrix = numeric\_df.corr()

# Plot heatmap  
 plt.figure(figsize=(8, 6))  
 sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm", fmt=".2f") plt.title("Correlation Heatmap for House Price Dataset")  
 plt.tight\_layout()  
 plt.show()

Columns in dataset: ['id', 'title', 'content', 'source', 'date\_published', 'labe

import matplotlib.pyplot as plt   
import numpy as np

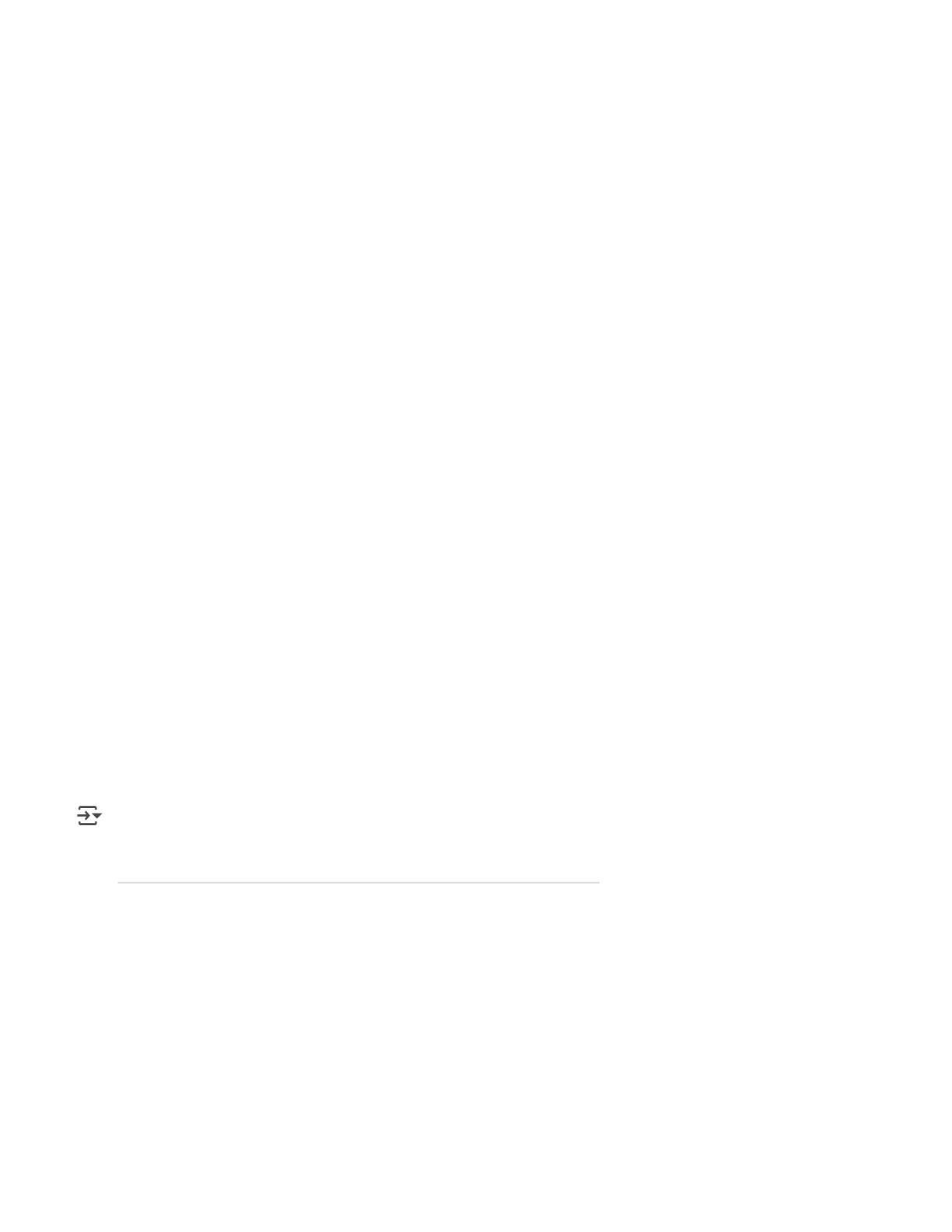
x = np.linspace(0, 10, 100)   
y = np.sin(x)

plt.plot(x,y,label="price")   
plt.xlabel("X Axis")   
plt.ylabel("Y Axis")   
plt.title("simple line plot")   
plt.legend()   
plt.show()

print(df.isnull().sum())   
df = df.dropna(subset=['label'])

# Assume 'sia' and 'label' are defined and accessible in this context # For example:   
# from nltk.sentiment.vader import SentimentIntensityAnalyzer   
# sia = SentimentIntensityAnalyzer()   
# label = "some text"

id 0   
title 0   
content 0   
source 0

date\_published 0   
label 0   
prediction 0   
confidence\_score 0   
dtype: int64

import pandas as pd   
from nltk.sentiment import SentimentIntensityAnalyzer import nltk

# Ensure VADER lexicon is downloaded   
nltk.download('vader\_lexicon')

# Load dataset   
df = pd.read\_csv("fake\_news\_detection\_dataset.csv")

# Drop rows with missing content   
df = df.dropna(subset=['content'])

# Initialize VADER sentiment analyzer   
sia = SentimentIntensityAnalyzer()

# Define emotion classifier   
def get\_emotion(text):  
 score = sia.polarity\_scores(text)['compound'] if score >= 0.05:  
 return "Positive"  
 elif score <= -0.05:  
 return "Negative"  
 else:  
 return "Neutral"

# Apply emotion classifier   
df['Emotion'] = df['content'].apply(get\_emotion)

# Display the updated DataFrame   
df[['title', 'Emotion']].head()

[nltk\_data] Downloading package vader\_lexicon to /root/nltk\_data... [nltk\_data] Package vader\_lexicon is already up-to-date!

title Emotion

|  |  |  |
| --- | --- | --- |
| 0 | Government Launches New Healthcare Program | Positive |

|  |  |  |
| --- | --- | --- |
| 1 | Aliens Land in Central Park and Offer Free Energy | Positive |

|  |  |  |
| --- | --- | --- |
| 2 | Tech Giant Acquires Startup to Boost AI Capabi... | Neutral |

|  |  |  |
| --- | --- | --- |
| 3 | Chocolate Cures All Diseases, Scientists Say | Neutral |