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

Choose Files Fake News …Dataset.csv

- **Fake News Detection Dataset.csv**(text/csv) - 127137 bytes, last modified: 5/20/2025 - 100% done

```
Saving Fake News Detection Dataset.csv to Fake News Detection Dataset.csv
```

```
import pandas as pd

df = pd.read_csv('Fake News Detection Dataset.csv')
```

```
print(df.head())
print("Shape:", df.shape)
print("Columns:", df.columns.tolist())
df.info()
df.describe(include='all')
```

```
     ID  Word_Count  Number_of_Sentence  Unique_Words  Average_Word_Length  \
0  1606          10                   4            24             6.176750
1  3718          10                   8            25             5.826770
2  2634          10                   7            18             4.619040
3  5560          10                   6            18             4.961424
4  7494          10                   4            21             4.114324

   Label
0      1
1      1
2      1
3      1
4      1
Shape: (4500, 6)
Columns: ['ID', 'Word_Count', 'Number_of_Sentence', 'Unique_Words', 'Average_Word_Ler
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4500 entries, 0 to 4499
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   4500 non-null   int64
 1   Word_Count           4500 non-null   int64
 2   Number_of_Sentence   4500 non-null   int64
 3   Unique_Words         4500 non-null   int64
 4   Average_Word_Length  4500 non-null   float64
 5   Label                4500 non-null   int64
dtypes: float64(1), int64(5)
memory usage: 211.1 KB
```

1 to 8 of 8 entries Filter

index	ID	Word_Count	Number_of_Sentence	Unique_Words	Averag
count	4500.0	4500.0	4500.0	4500.0	
mean	5469.14	53.934	8.934666666666667	24.943333333333333	4.9
std	2599.1930594945406	24.87274300802541	3.407847330416677	11.540708027016231	1.15
min	1002.0	10.0	4.0	5.0	
25%	3228.75	35.0	6.0	17.0	
50%	5449.5	52.0	9.0	22.0	
75%	7706.75	75.0	12.0	33.0	
max	9999.0	100.0	15.0	50.0	

Show 10 per page

```
print("Missing values:\n", df.isnull().sum())
print("Duplicate rows:", df.duplicated().sum())
```

```
Missing values:
 ID                     0
Word_Count             0
Number_of_Sentence     0
Unique_Words           0
Average_Word_Length    0
Label                  0
dtype: int64
```

```
Duplicate rows: 0
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# Class distribution (assuming 'label' column: 1 = fake, 0 = real)
sns.countplot(data=df, x='label')
plt.title('Distribution of Fake and Real News')
plt.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-7-51a749c26884> in <cell line: 0>()
      3 
      4 # Class distribution (assuming 'label' column: 1 = fake, 0 = real)
----> 5 sns.countplot(data=df, x='label')
      6 plt.title('Distribution of Fake and Real News')
      7 plt.show()

5 frames

/usr/local/lib/python3.11/dist-packages/seaborn/_core/data.py in
_assign_variables(self, data, variables)
    230                 else:
    231                     err += "An entry with this name does not appear in
`data`."
--> 232                 raise ValueError(err)
    233 
    234             else:

ValueError: Could not interpret value `label` for `x`. An entry with this name does
```

Next steps: Explain error

```
X = df['text']  # Assuming 'text' is the news content
y = df['label']  # 0 = real, 1 = fake
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in get_loc(self,
key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
   3806         except KeyError as err:

index.pyx in pandas._libs.index.IndexEngine.get_loc()

index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'text'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
                              2 frames
/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in get_loc(self,
key)
   3810             ):
   3811                 raise InvalidIndexError(key)
-> 3812             raise KeyError(key) from err
   3813         except TypeError:
   3814             # If we have a listlike key, _check_indexing_error will raise
```

Next steps: Explain error

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(X_train_vec, y_train)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-9-daa24eab425f> in <cell line: 0>()
      2
      3 model = LogisticRegression()
----> 4 model.fit(X_train_vec, y_train)

NameError: name 'X_train_vec' is not defined
```

Next steps: Explain error

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

y_pred = model.predict(X_test_vec)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```python
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-10-2a674300417e> in <cell line: 0>()
      1 from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
      2 
----> 3 y_pred = model.predict(X_test_vec)
      4 print("Accuracy:", accuracy_score(y_test, y_pred))
      5 print(confusion_matrix(y_test, y_pred))

NameError: name 'X_test_vec' is not defined
```

Next steps: Explain error

```python
sample_news = ["This is a sample news article text..."]
sample_vec = vectorizer.transform(sample_news)
pred = model.predict(sample_vec)

print("Prediction (1 = Fake, 0 = Real):", pred[0])
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-11-50c538ec0fbb> in <cell line: 0>()
      1 sample_news = ["This is a sample news article text..."]
----> 2 sample_vec = vectorizer.transform(sample_news)
      3 pred = model.predict(sample_vec)
      4 
      5 print("Prediction (1 = Fake, 0 = Real):", pred[0])

NameError: name 'vectorizer' is not defined
```

Next steps: Explain error

```python
import gradio as gr

def predict_fake_news(news_text):
    vec = vectorizer.transform([news_text])
    prediction = model.predict(vec)[0]
    return "Fake News" if prediction == 1 else "Real News"

gr.Interface(
    fn=predict_fake_news,
    inputs=gr.Textbox(lines=10, placeholder="Paste news article here..."),
    outputs="text",
    title="📰 Fake News Detector",
    description="Paste a news article and get prediction: Real or Fake"
).launch()
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-12-7f30a1c75034> in <cell line: 0>()
----> 1 import gradio as gr
      2 
      3 def predict_fake_news(news_text):
      4     vec = vectorizer.transform([news_text])
      5     prediction = model.predict(vec)[0]

ModuleNotFoundError: No module named 'gradio'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```

OPEN EXAMPLES

Next steps: Explain error

```
!pip install gradio
```

```
Collecting python-multipart>=0.0.18 (from gradio)
  Downloading python_multipart-0.0.20-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-p
Collecting ruff>=0.9.3 (from gradio)
  Downloading ruff-0.11.10-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.
Collecting safehttpx<0.2.0,>=0.1.6 (from gradio)
```

```
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.1
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.1
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-package
Downloading gradio-5.30.0-py3-none-any.whl (54.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.2/54.2 MB 19.4 MB/s eta 0:00:00
Downloading gradio_client-1.10.1-py3-none-any.whl (323 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 323.1/323.1 kB 25.7 MB/s eta 0:00:00
Downloading aiofiles-24.1.0-py3-none-any.whl (15 kB)
Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 95.2/95.2 kB 8.8 MB/s eta 0:00:00
Downloading groovy-0.1.2-py3-none-any.whl (14 kB)
Downloading python_multipart-0.0.20-py3-none-any.whl (24 kB)
Downloading ruff-0.11.10-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.6/11.6 MB 102.3 MB/s eta 0:00:00
Downloading safehttpx-0.1.6-py3-none-any.whl (8.7 kB)
Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Downloading starlette-0.46.2-py3-none-any.whl (72 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 72.0/72.0 kB 6.9 MB/s eta 0:00:00
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
```

```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression

# Vectorize text
vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
X_vec = vectorizer.fit_transform(X)

# Train model
model = LogisticRegression()
model.fit(X_vec, y)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-14-46f1f5175d65> in <cell line: 0>()
      4 # Vectorize text
      5 vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
----> 6 X_vec = vectorizer.fit_transform(X)
      7
      8 # Train model

NameError: name 'X' is not defined
```

Next steps: Explain error

```python
import gradio as gr

def predict_news(news_text):
    vec = vectorizer.transform([news_text])
    pred = model.predict(vec)[0]
    return "Fake News" if pred == 1 else "Real News"



gr.Interface(
    fn=predict_news,
    inputs=gr.Textbox(lines=10, placeholder="Paste news article here..."),
    outputs="text",
    title="📰 Fake News Detector",
    description="Enter a news article below to check if it's real or fake."
).launch(share=True)  # share=True creates a public URL
```

```
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://4a48acb500c9b7f2cf.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `
```

📰 Fake News Detector

Enter a news article below to check if it's real or fake.

news_text

Paste news article here...