1. Introduction to Fake News Detection

Fake news refers to false or misleading information presented as real news. With the rapid growth of social media and online platforms, fake news spreads quickly and can influence public opinion, politics, and social behavior. Manual verification of news is time-consuming and inefficient, hence automated fake news detection using Natural Language Processing (NLP) and Machine Learning (ML) has become an important research area.

This project focuses on identifying whether a given news statement is Real or Fake using text-based machine learning techniques.

2. Problem Statement

To design and implement a system that can automatically classify news articles as Real or Fake based on their textual content using machine learning algorithms.

In [2]:
```
# 1. Import Libraries
import pandas as pd
import numpy as np
import re
import pickle
```

In [3]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, KFold
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix, classification_r
```

In [4]:
```
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
```

In [5]:
```
# Download required NLTK resources (run once)
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\yj372\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\yj372\AppData\Roaming\nltk_data...
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\yj372\AppData\Roaming\nltk_data...
```

Out[5]:
```
True
```

3. Dataset Description

The dataset used in this project is derived from the LIAR Dataset, which is a benchmark dataset for fake news detection.

Dataset Files: train.csv

test.csv

valid.csv

Selected Features: Statement: News headline or text

Label: Binary class (True / False)

To simplify classification, the original multi-class labels were converted into two classes:

True → Real News

False → Fake News

In [7]:

```
# ============================================
# 2. Load Dataset
# ============================================
train_df = pd.read_csv("train.csv")
test_df  = pd.read_csv("test.csv")
valid_df = pd.read_csv("valid.csv")
```

In [8]:

```
# Combine all datasets for simplicity
data = pd.concat([train_df, test_df, valid_df], axis=0).reset_index(drop=True)
```

In [9]:

```
print("Dataset Shape:", data.shape)
print(data.head())
```

```
Dataset Shape: (15362, 2)
                                           Statement  Label
0  Says the Annies List political group supports ...  False
1  When did the decline of coal start? It started...   True
2  Hillary Clinton agrees with John McCain "by vo...   True
3  Health care reform legislation is likely to ma...  False
4  The economic turnaround started at the end of ...   True
```

4. Text Preprocessing

Raw text data contains noise such as punctuation, stopwords, and irrelevant symbols. Preprocessing improves the quality of data before model training.

Preprocessing Steps: Convert text to lowercase

Remove punctuation and special characters

Tokenization (splitting text into words)

Stopword removal (e.g., “is”, “the”, “and”)

Lemmatization (reducing words to base form)

These steps help reduce dimensionality and improve model performance.

In [34]:
```
stop_words = set(stopwords.words("english"))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    text = re.sub(r'[^a-zA-Z]', ' ', str(text))
    text = text.lower()
    words = text.split()
    words = [lemmatizer.lemmatize(word) for word in words if word not in stop_words]
    return " ".join(words)

data["clean_text"] = data["Statement"].apply(preprocess_text)
```

5. Feature Extraction

Machine learning models cannot work directly on text, so text must be converted into numerical features.

5.1 TF-IDF (Term Frequency – Inverse Document Frequency) TF-IDF assigns importance to words based on:

How frequently a word appears in a document (TF)

How rare the word is across all documents (IDF)

Words that are frequent in a document but rare globally receive higher weights, making TF-IDF effective for text classification.

In [11]:
```
# ================================================
# 4. Feature Extraction (TF-IDF)
# ================================================
X = data["clean_text"]
y = data["Label"]

tfidf = TfidfVectorizer(ngram_range=(1,2), max_df=0.7)
X_tfidf = tfidf.fit_transform(X)
```

In [14]:
```
# Remove rows with missing values
data = data.dropna(subset=["Statement", "Label"]).reset_index(drop=True)
```

In [15]:
```
# Convert labels to numeric if needed
data["Label"] = data["Label"].map({
    "True": 1,
    "False": 0,
    1: 1,
```

```
    0: 0
})
```

In [16]:

```
data = data.dropna(subset=["Label"]).reset_index(drop=True)
data["Label"] = data["Label"].astype(int)
```

In [17]:

```
def preprocess_text(text):
    if pd.isna(text):
        return ""
    text = re.sub(r'[^a-zA-Z]', ' ', text)
    text = text.lower()
    words = text.split()
    words = [lemmatizer.lemmatize(word) for word in words if word not in stop_words]
    return " ".join(words)
```

In [18]:

```
data["clean_text"] = data["Statement"].apply(preprocess_text)

# Remove empty cleaned texts
data = data[data["clean_text"].str.strip() != ""].reset_index(drop=True)
```

In [19]:

```
X = data["clean_text"]
y = data["Label"]

tfidf = TfidfVectorizer(ngram_range=(1,2), max_df=0.7)
X_tfidf = tfidf.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(
    X_tfidf,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
```

6. Machine Learning Algorithm Used

Logistic Regression Logistic Regression is a supervised learning algorithm used for binary classification problems. It estimates the probability that a given input belongs to a particular class using a sigmoid function.

Reasons for choosing Logistic Regression:

Works well with high-dimensional sparse data

Efficient and easy to interpret

Performs well with TF-IDF features

In [20]:

```
# ============================================
# 6. Model Training (Logistic Regression)
```

```
# ============================================
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

Out[20]:

▾ LogisticRegression

▶ Parameters

7. Model Training and Evaluation

The dataset is split into training and testing sets. The model is trained on the training data and evaluated on unseen test data.

Evaluation Metrics: Accuracy: Overall correctness of the model

Precision: Correct positive predictions

Recall: Ability to detect all positive cases

F1-Score: Balance between precision and recall

Confusion Matrix: Shows correct and incorrect predictions

K-Fold Cross Validation is used to ensure model stability and reduce overfitting.

In [21]:

```
# ============================================
# 7. Model Evaluation
# ============================================
y_pred = model.predict(X_test)

print("\nAccuracy:", accuracy_score(y_test, y_pred))
print("F1 Score:", f1_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.6205549042594763
F1 Score: 0.7058467131172372

Confusion Matrix:
 [[ 423  709]
 [ 262 1165]]

Classification Report:
              precision    recall  f1-score   support

           0       0.62      0.37      0.47      1132
           1       0.62      0.82      0.71      1427

    accuracy                           0.62      2559
   macro avg       0.62      0.60      0.59      2559
weighted avg       0.62      0.62      0.60      2559
```

In [22]:

```
# ================================================
# 8. K-Fold Cross Validation
# ================================================
kf = KFold(n_splits=5, shuffle=True, random_state=42)
f1_scores = []

for train_idx, val_idx in kf.split(X_tfidf):
    X_tr, X_val = X_tfidf[train_idx], X_tfidf[val_idx]
    y_tr, y_val = y.iloc[train_idx], y.iloc[val_idx]

    clf = LogisticRegression(max_iter=1000)
    clf.fit(X_tr, y_tr)
    preds = clf.predict(X_val)
    f1_scores.append(f1_score(y_val, preds))

print("Average K-Fold F1 Score:", np.mean(f1_scores))
```

```
Average K-Fold F1 Score: 0.6978933379244794
```

In [23]:

```
# ================================================
# 9. Save Trained Model
# ================================================
pickle.dump(model, open("final_model.sav", "wb"))
pickle.dump(tfidf, open("tfidf_vectorizer.pkl", "wb"))

print("Model and Vectorizer saved successfully.")
```

```
Model and Vectorizer saved successfully.
```

8. Prediction Mechanism

The trained model predicts whether a news statement is Real or Fake by:

Preprocessing the input text

Converting it into TF-IDF vectors

Passing it through the trained model

Producing a class label with probability score

The model uses probability thresholds to reduce incorrect predictions.

In [24]:

```
# ================================================
# 10. Prediction Function
# ================================================
def predict_news(news_text):
    cleaned = preprocess_text(news_text)
    vectorized = tfidf.transform([cleaned])
    prediction = model.predict(vectorized)[0]

    if prediction == 1:
        return "Real News 📰"
```

```
    else:
        return "Fake News ⚠"
```

In [25]:

```
# ================================================
# 11. Sample Prediction
# ================================================
sample_news = "The government has announced a new policy to improve education."
print("\nSample Prediction:", predict_news(sample_news))
```

```
Sample Prediction: Fake News ⚠
```

In [33]:

```
test_news_samples = [

    "The government announced a new education policy aimed at improving digital literacy

    "The Reserve Bank of India decided to keep interest rates unchanged after reviewing

    "Scientists have discovered a new species of frog in the Amazon rainforest, accordin

    "The Indian Space Research Organisation successfully launched a new communication sa

    "The Supreme Court issued a new directive regarding environmental protection laws.",

    "Breaking: Government secretly plans to ban all social media platforms starting tomo

    "Scientists confirm the Earth will stop rotating for five days next month, causing g

    "Drinking salt water every morning can completely cure cancer, doctors say.",

    "Aliens have contacted world leaders and will reveal themselves on live television t

    "A viral message claims that using mobile phones after midnight causes instant brain

    "Experts warn that excessive screen time may negatively impact mental health if not

    "A new study suggests that coffee consumption could be linked to improved focus and

    "Unverified sources claim a major economic announcement will be made soon.",

    "Health officials are investigating reports related to a newly emerging virus strain

    "Social media users are debating the effectiveness of a newly launched government sc
]

# Run predictions
for i, news in enumerate(test_news_samples, 1):
    print(f"{i}. {news}")
    print("   ➜ Prediction:", predict_news(news))
    print("-" * 80)
```

```
1. The government announced a new education policy aimed at improving digital literacy i
n rural areas.
   ➜ Prediction: Fake News ⚠ (confidence: 0.60)
--------------------------------------------------------------------------------
2. The Reserve Bank of India decided to keep interest rates unchanged after reviewing cu
```

```
rrent inflation data.
   → Prediction: Fake News ⚠ (confidence: 0.48)
--------------------------------------------------------------------------------
3. Scientists have discovered a new species of frog in the Amazon rainforest, according
to a peer-reviewed study.
   → Prediction: Fake News ⚠ (confidence: 0.42)
--------------------------------------------------------------------------------
4. The Indian Space Research Organisation successfully launched a new communication sate
llite into orbit.
   → Prediction: Real News 📰 (confidence: 0.62)
--------------------------------------------------------------------------------
5. The Supreme Court issued a new directive regarding environmental protection laws.
   → Prediction: Fake News ⚠ (confidence: 0.53)
--------------------------------------------------------------------------------
6. Breaking: Government secretly plans to ban all social media platforms starting tomorr
ow!
   → Prediction: Fake News ⚠ (confidence: 0.57)
--------------------------------------------------------------------------------
7. Scientists confirm the Earth will stop rotating for five days next month, causing glo
bal chaos.
   → Prediction: Fake News ⚠ (confidence: 0.44)
--------------------------------------------------------------------------------
8. Drinking salt water every morning can completely cure cancer, doctors say.
   → Prediction: Fake News ⚠ (confidence: 0.59)
--------------------------------------------------------------------------------
9. Aliens have contacted world leaders and will reveal themselves on live television ton
ight.
   → Prediction: Fake News ⚠ (confidence: 0.43)
--------------------------------------------------------------------------------
10. A viral message claims that using mobile phones after midnight causes instant brain
damage.
   → Prediction: Fake News ⚠ (confidence: 0.56)
--------------------------------------------------------------------------------
11. Experts warn that excessive screen time may negatively impact mental health if not m
anaged properly.
   → Prediction: Real News 📰 (confidence: 0.63)
--------------------------------------------------------------------------------
12. A new study suggests that coffee consumption could be linked to improved focus and p
roductivity.
   → Prediction: Fake News ⚠ (confidence: 0.44)
--------------------------------------------------------------------------------
13. Unverified sources claim a major economic announcement will be made soon.
   → Prediction: Real News 📰 (confidence: 0.61)
--------------------------------------------------------------------------------
14. Health officials are investigating reports related to a newly emerging virus strain.
   → Prediction: Fake News ⚠ (confidence: 0.55)
--------------------------------------------------------------------------------
15. Social media users are debating the effectiveness of a newly launched government sch
eme.
   → Prediction: Fake News ⚠ (confidence: 0.69)
--------------------------------------------------------------------------------
```

In [35]:

```python
while True:
    text = input("\nEnter news text (or type 'exit'): ")
    if text.lower() == "exit":
        break
    print("Prediction:", predict_news(text))
```

9. Limitations of the System

The model is pattern-based, not fact-based

It does not verify news from real-time sources

Some true statements may be misclassified due to dataset bias

Performance depends on the quality and size of the dataset

10. Applications

Social media monitoring

News credibility analysis

Content moderation systems

Journalism assistance tools

11. Future Scope

Use deep learning models such as LSTM or BERT

Incorporate source credibility and metadata

Train on larger and more diverse datasets

Integrate real-time fact-checking APIs

12. Conclusion

This project demonstrates the use of Natural Language Processing and Machine Learning techniques to detect fake news effectively. By applying text preprocessing, TF-IDF feature extraction, and Logistic Regression, the system achieves reasonable accuracy and provides a foundation for more advanced fake news detection systems

In [ ]: