```
# Fake News Detection using NLP

# Step 1: Install Required Libraries
# pip install pandas numpy scikit-learn matplotlib seaborn

# Step 2: Load Dataset
import pandas as pd
df_fake = pd.read_csv("Fake.csv")
df_fake['label'] = 0

df_real = pd.read_csv("True.csv")
df_real['label'] = 1

data = pd.concat([df_fake, df_real])
data = data.sample(frac=1).reset_index(drop=True)

# Step 3: Preprocessing
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

X = data['text']
y = data['label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Step 4: Train Model
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

model = LogisticRegression()
model.fit(X_train_tfidf, y_train)

y_pred = model.predict(X_test_tfidf)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

# Step 5: Predict New News
def predict_news(news_text):
    vec = vectorizer.transform([news_text])
    pred = model.predict(vec)[0]
    conf = model.predict_proba(vec)[0].max()
    return "Real" if pred == 1 else "Fake", conf

news = "NASA discovers alien life on Mars"
label, confidence = predict_news(news)
print(f"Prediction: {label} (Confidence: {confidence:.2f})")
```