```
#Import necessary libraries
import pandas as pd
import numpy as np
import re
import string
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from sklearn.model_selection import cross_val_score
import matplotlib.pyplot as plt
import seaborn as sns

#Load the dataset
data = pd.read_csv("/content/drive/MyDrive/fake_and_real_news.csv")

#Explore the data
print(data.head())
print(data.columns)

#Map 'Fake' to 0 and 'Real' to 1
data['class'] = data['label'].map({'Fake': 0, 'Real': 1})

#Split the data into features and labels
x = data['Text']  # Assuming the column name is 'Text'
y = data['class']

#Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)

#Define a function to clean the text data
def clean_text(text):
    text = text.lower()
    text = re.sub(r'https?://\S+|www\.\S+', '', text)  # remove URLs
    text = re.sub(r'[^a-z\s]', '', text)  # remove punctuation and numbers
    text = re.sub(r'\s+', ' ', text).strip()  # remove extra spaces
    return text

#Apply the cleaning function to the text data
x_train = x_train.apply(clean_text)
x_test = x_test.apply(clean_text)

#Create a TF-IDF vectorizer
vectorizer = TfidfVectorizer(ngram_range=(1,2), max_df=0.8, min_df=5)

#Fit the vectorizer to the training data and transform both the training and testing data
xv_train = vectorizer.fit_transform(x_train)
xv_test = vectorizer.transform(x_test)

#Define the machine learning models with tuned hyperparameters
models = {
    'Logistic Regression': LogisticRegression(max_iter=10000, C=1.0),
    'Decision Tree Classifier': DecisionTreeClassifier(max_depth=5),
    'Random Forest Classifier': RandomForestClassifier(n_estimators=100, max_depth=5),
    'Gradient Boosting Classifier': GradientBoostingClassifier(n_estimators=100, learning_rate=0.1)
}

#Train and evaluate each model, storing results
model_results = {}
for name, model in models.items():
    model.fit(xv_train, y_train)
    y_pred = model.predict(xv_test)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    model_results[name] = {'accuracy': accuracy, 'precision': precision, 'recall': recall, 'f1': f1}
    print(f"Model: {name}")
    print(f"Accuracy: {accuracy:.2f}")
    print(f"Precision: {precision:.2f}")
    print(f"Recall: {recall:.2f}")
    print(f"F1-score: {f1:.2f}")
    cm = confusion_matrix(y_test, y_pred)
```

```
        plt.figure(figsize=(6,4))
        sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
        plt.title(f"Confusion Matrix for {name}")
        plt.xlabel("Predicted")
        plt.ylabel("Actual")
        plt.show()

    #Select the best model based on F1-score
    best_model_name = max(model_results, key=lambda x: model_results[x]['f1'])
    best_model = models[best_model_name]

    #Use cross-validation to evaluate the model's performance
    scores = cross_val_score(best_model, xv_train, y_train, cv=3, scoring='f1_macro')
    print(f"Best Model: {best_model_name}")
    print(f"Cross-validated F1-score: {scores.mean():.2f}")

    #Define a function to predict news
    def predict_news(news):
        cleaned = clean_text(news)
        vect_news = vectorizer.transform([cleaned])
        pred = best_model.predict(vect_news)
        label = "Real" if pred[0] == 1 else "Fake"
        return label

    #Test the prediction function
    news = "BUSTED: Trump Supporter Used Poll Watcher Credentials To Force Early Voters To Leave."
    print(predict_news(news))
```

```
                                                Text label
0   Top Trump Surrogate BRUTALLY Stabs Him In The...  Fake
1  U.S. conservative leader optimistic of common ...  Real
2  Trump proposes U.S. tax overhaul, stirs concer...  Real
3   Court Forces Ohio To Allow Millions Of Illega...  Fake
4  Democrats say Trump agrees to work on immigrat...  Real
Index(['Text', 'label'], dtype='object')
Model: Logistic Regression
Accuracy: 0.99
Precision: 1.00
Recall: 0.99
F1-score: 0.99
```