```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split,
RandomizedSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier,
AdaBoostClassifier
from xgboost import XGBClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
from google.colab import drive

# Mount Google Drive
drive.mount('/content/drive')

# Load dataset
df = pd.read_csv('/content/drive/MyDrive/fake_and_real_news.csv')

# Drop NaN values
df.dropna(inplace=True)

# Encode labels
label_encoder = LabelEncoder()
df['label'] = label_encoder.fit_transform(df['label'])  # Converts
'Fake'/'Real' to 0/1

# Splitting dataset
X = df['Text']
y = df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Vectorization
tfidf = TfidfVectorizer(stop_words='english', max_features=5000)
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)

# Define classifiers
classifiers = {
    "Naive Bayes": MultinomialNB(),
    "SVM": SVC(probability=True),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(),
    "AdaBoost": AdaBoostClassifier(),
    "XGBoost": XGBClassifier(use_label_encoder=False,
eval_metric='logloss'),
    "MLP": MLPClassifier(max_iter=300)
}

# Hyperparameter tuning setup
param_grid = {
    "SVM": {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf']},
    "Decision Tree": {'max_depth': [10, 20, 30]},
    "Random Forest": {'n_estimators': [50, 100, 200]},
    "MLP": {'hidden_layer_sizes': [(50,), (100,), (50,50)],
'activation': ['relu', 'tanh']}
}

# Train and evaluate models
results = {}
for name, model in classifiers.items():
    print(f"Training {name}...")
    if name in param_grid:
        search = RandomizedSearchCV(model, param_grid[name], n_iter=5,
cv=3, scoring='accuracy', verbose=0, n_jobs=-1)
        search.fit(X_train_tfidf, y_train)
        model = search.best_estimator_
    else:
        model.fit(X_train_tfidf, y_train)

    y_pred = model.predict(X_test_tfidf)
    acc = accuracy_score(y_test, y_pred)
    print(f"{name} Accuracy: {acc:.4f}")
    print(classification_report(y_test, y_pred))
    results[name] = acc

# Display results
print("\nFinal Results:")
for model, acc in results.items():
    print(f"{model}: {acc:.4f}")
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
Training Naive Bayes...
Naive Bayes Accuracy: 0.9636
              precision    recall  f1-score   support

           0       0.97      0.96      0.96       973
           1       0.96      0.97      0.96      1007

    accuracy                           0.96      1980
   macro avg       0.96      0.96      0.96      1980
weighted avg       0.96      0.96      0.96      1980

Training SVM...
```

```
SVM Accuracy: 0.9970
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       973
           1       1.00      1.00      1.00      1007

    accuracy                           1.00      1980
   macro avg       1.00      1.00      1.00      1980
weighted avg       1.00      1.00      1.00      1980

Training Decision Tree...

/usr/local/lib/python3.11/dist-packages/sklearn/model_selection/
_search.py:317: UserWarning: The total space of parameters 3 is
smaller than n_iter=5. Running 3 iterations. For exhaustive searches,
use GridSearchCV.
  warnings.warn(

Decision Tree Accuracy: 0.9985
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       973
           1       1.00      1.00      1.00      1007

    accuracy                           1.00      1980
   macro avg       1.00      1.00      1.00      1980
weighted avg       1.00      1.00      1.00      1980

Training Random Forest...

/usr/local/lib/python3.11/dist-packages/sklearn/model_selection/
_search.py:317: UserWarning: The total space of parameters 3 is
smaller than n_iter=5. Running 3 iterations. For exhaustive searches,
use GridSearchCV.
  warnings.warn(

Random Forest Accuracy: 0.9990
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       973
           1       1.00      1.00      1.00      1007

    accuracy                           1.00      1980
   macro avg       1.00      1.00      1.00      1980
weighted avg       1.00      1.00      1.00      1980

Training AdaBoost...
AdaBoost Accuracy: 0.9995
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       973
           1       1.00      1.00      1.00      1007

    accuracy                           1.00      1980
   macro avg       1.00      1.00      1.00      1980
weighted avg       1.00      1.00      1.00      1980

Training XGBoost...

/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158:
UserWarning: [06:58:09] WARNING: /workspace/src/learner.cc:740:
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)

XGBoost Accuracy: 0.9995
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       973
           1       1.00      1.00      1.00      1007

    accuracy                           1.00      1980
   macro avg       1.00      1.00      1.00      1980
weighted avg       1.00      1.00      1.00      1980

Training MLP...
MLP Accuracy: 0.9944
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       973
           1       0.99      1.00      0.99      1007

    accuracy                           0.99      1980
   macro avg       0.99      0.99      0.99      1980
weighted avg       0.99      0.99      0.99      1980


Final Results:
Naive Bayes: 0.9636
SVM: 0.9970
Decision Tree: 0.9985
Random Forest: 0.9990
AdaBoost: 0.9995
XGBoost: 0.9995
MLP: 0.9944
```

```
!pip uninstall -y numpy joblib scikit-learn
!pip install numpy==1.26.4 joblib scikit-learn
```

```
Found existing installation: numpy 1.26.4
Uninstalling numpy-1.26.4:
  Successfully uninstalled numpy-1.26.4
```

```
Found existing installation: joblib 1.4.2
Uninstalling joblib-1.4.2:
  Successfully uninstalled joblib-1.4.2
Found existing installation: scikit-learn 1.6.1
Uninstalling scikit-learn-1.6.1:
  Successfully uninstalled scikit-learn-1.6.1
Collecting numpy==1.26.4
  Using cached numpy-1.26.4-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (61 kB)
Collecting joblib
  Using cached joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Collecting scikit-learn
  Using cached scikit_learn-1.6.1-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (18 kB)
Requirement already satisfied: scipy>=1.6.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.14.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Using cached numpy-1.26.4-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.3 MB)
Using cached joblib-1.4.2-py3-none-any.whl (301 kB)
Using cached scikit_learn-1.6.1-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.5 MB)
Installing collected packages: numpy, joblib, scikit-learn
Successfully installed joblib-1.4.2 numpy-1.26.4 scikit-learn-1.6.1
```