Fake News Detection

Task 1

```
pip install colorama
```

```
Requirement already satisfied: colorama in c:\python310\lib\site-packages (0.4.6)
Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 23.0 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
# Load the datasets
fake_train = pd.read_csv('Fake_train.csv')
fake_dev = pd.read_csv('Fake_dev.csv')
```

```
# Vectorize the text using TF-IDF
vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform(fake_train['text'])
y_train = fake_train['label']
```

```
X_dev = vectorizer.transform(fake_dev['text'])
y_dev = fake_dev['label']
```

Naive Bayes

```
# Initialize a Naive Bayes classifier
classifier_nb = MultinomialNB()
classifier_nb.fit(X_train, y_train)
```

```
MultinomialNB
MultinomialNB()
```

```
# Predict on the development set
predictions_nb = classifier_nb.predict(X_dev)
```

```
results_nb = pd.DataFrame({
    'Text': fake_dev['text'],
    'Actual': y_dev,
    'Predicted': predictions_nb
})
```

```
# Calculate the accuracy
accuracy_nb = accuracy_score(y_dev, predictions_nb)
print(f"Accuracy: {accuracy_nb}")
```

```
Accuracy: 0.7889570552147239
```

```
from colorama import Fore, Style

print("Actual  Predicted")
for actual, predicted in zip(results_nb['Actual'], results_nb['Predicted']):
    if actual == predicted:
        print(f"{actual:<8} {predicted:<10}")
    else:
        print(f"{Fore.RED}{actual:<8}{Style.RESET_ALL} {Fore.RED}{predicted:<10}{Style.RESET_ALL}")
```

```
Actual  Predicted
Fake     Fake
```

```
Fake     Fake
original Fake
original original
original Fake
Fake     Fake
original Fake
original Fake
original Fake
Fake     Fake
Fake     Fake
original Fake
Fake     original
Fake     Fake
Fake     Fake
original original
original original
Fake     Fake
original original
Fake     Fake
original original
Fake     original
Fake     Fake
Fake     Fake
original original
original Fake
Fake     Fake
Fake     Fake
original original
original original
Fake     Fake
original original
original original
Fake     Fake
Fake     Fake
Fake     Fake
original original
original Fake
original original
Fake     Fake
Fake     Fake
original original
Fake     Fake
Fake     original
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
original original
original original
Fake     Fake
Fake     Fake
original original
original Fake
```

SVM

```
classifier_svm = SVC(kernel='rbf')
classifier_svm.fit(X_train, y_train)
```

SVC
SVC()

```
# Predict on the development set
predictions_svm = classifier_svm.predict(X_dev)
```

```
results_svm = pd.DataFrame({
    'Text': fake_dev['text'],
    'Actual': y_dev,
    'Predicted': predictions_svm
})
```

```
# Calculate the accuracy
accuracy_svm = accuracy_score(y_dev, predictions_svm)
print(f"Accuracy: {accuracy_svm}")
```

```
Accuracy: 0.7877300613496933
```

```
print("Actual  Predicted")
for actual, predicted in zip(results_svm['Actual'], results_svm['Predicted']):
    if actual == predicted:
        print(f"{actual:<8} {predicted:<10}")
    else:
        print(f"{Fore.RED}{actual:<8}{Style.RESET_ALL} {Fore.RED}{predicted:<10}{Style.RESET_ALL}")
```

```
Fake     Fake
Fake     Fake
original original
original original
Fake     Fake
original original
original original
Fake     Fake
Fake     Fake
Fake     Fake
original original
original Fake
original original
Fake     Fake
Fake     Fake
original original
Fake     original
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
original original
original original
Fake     Fake
Fake     Fake
original original
original original
original original
Fake     Fake
original original
original original
original original
Fake     original
original Fake
original original
Fake     Fake
Fake     Fake
original Fake
original original
Fake     Fake
Fake     Fake
original original
original original
original original
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
original Fake
Fake     original
Fake     Fake
original original
original original
Fake     original
original Fake
original original
```

Random Forest

```
param_grid = {'n_estimators': [50, 100, 200],
              'max_depth': [None, 10, 20, 30],
              'min_samples_split': [2, 5, 10],
              'min_samples_leaf': [1, 2, 4]}
```

```
classifier = RandomForestClassifier(random_state=42)
grid_search = GridSearchCV(classifier, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
```

```
GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier
```

```
# Predict on the development set
best_classifier_rf = grid_search.best_estimator_
predictions_rf = best_classifier_rf.predict(X_dev)
```

```
results_rf = pd.DataFrame({
    'Text': fake_dev['text'],
    'Actual': y_dev,
    'Predicted': predictions_rf
})
```

```
# Calculate the accuracy
accuracy_rf = accuracy_score(y_dev, predictions_rf)
print(f"Accuracy: {accuracy_rf}")
```

```
Accuracy: 0.7607361963190185
```

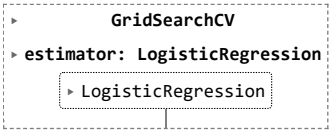
```
print("Actual  Predicted")
for actual, predicted in zip(results_rf['Actual'], results_rf['Predicted']):
    if actual == predicted:
        print(f"{actual:<8} {predicted:<10}")
    else:
        print(f"{Fore.RED}{actual:<8}{Style.RESET_ALL} {Fore.RED}{predicted:<10}{Style.RESET_ALL}")
```

```
original original
original original
Fake     Fake
Fake     Fake
original original
original original
original original
Fake     Fake
Fake     Fake
Fake     Fake
Fake     Fake
original original
Fake     original
Fake     Fake
original original
original original
Fake     original
original Fake
original original
original original
original original
Fake     Fake
Fake     Fake
Fake     original
Fake     original
Fake     original
original original
original original
Fake     original
Fake     original
Fake     original
original original
original original
Fake     Fake
original original
Fake     original
original original
Fake     Fake
Fake     Fake
Fake     Fake
original original
Fake     original
Fake     original
Fake     original
Fake     original
Fake     Fake
original Fake
original original
original original
Fake     Fake
original original
original original
original original
original original
Fake     Fake
original original
Fake     Fake
original original
Fake     Fake
```

Logistic Regression

```
param_grid_lr = {'C': [0.001, 0.01, 0.1, 1, 10, 100],
              'solver': ['liblinear', 'lbfgs', 'newton-cg', 'sag', 'saga']}
```

```
classifier_lr = LogisticRegression(max_iter=1000, random_state=42)
grid_search_lr = GridSearchCV(classifier_lr, param_grid_lr, cv=5, scoring='accuracy', n_jobs=-1)
grid_search_lr.fit(X_train, y_train)
```




```
# Predict on the development set
best_classifier_lr = grid_search_lr.best_estimator_
predictions_lr = best_classifier_lr.predict(X_dev)
```

```
results_lr = pd.DataFrame({
    'Text': fake_dev['text'],
    'Actual': y_dev,
    'Predicted': predictions_lr
})
```

```
# Calculate the accuracy
accuracy_lr = accuracy_score(y_dev, predictions_lr)
print(f"Accuracy: {accuracy_lr}")
```

```
Accuracy: 0.7865030674846626
```

```
print("Actual  Predicted")
for actual, predicted in zip(results_lr['Actual'], results_lr['Predicted']):
    if actual == predicted:
        print(f"{actual:<8} {predicted:<10}")
    else:
        print(f"{Fore.RED}{actual:<8}{Style.RESET_ALL} {Fore.RED}{predicted:<10}{Style.RESET_ALL}")
```

```
Fake     Fake
original original
original original
original original
original original
Fake     Fake
original original
Fake     Fake
original original
Fake     Fake
original Fake
```

Inference

```
print("NB Accuracy: ",accuracy_nb)
print("NB Accuracy: ",accuracy_svm)
print("NB Accuracy: ",accuracy_rf)
print("NB Accuracy: ",accuracy_lr)
```

```
NB Accuracy:  0.7889570552147239
NB Accuracy:  0.7877300613496933
NB Accuracy:  0.7607361963190185
NB Accuracy:  0.7865030674846626
```

Applying NB in test data which is without label.

```
test_without_label = pd.read_csv('Fake_test_without_labels.csv')
```

```
x = vectorizer.transform(test_without_label['text'])
```

```
predictions_without_label_nb = classifier_nb.predict(x)
```

```
results_without_label_nb = pd.DataFrame({
    'Text': test_without_label['text'],
    'Predicted': predictions_without_label_nb
})
```

```
results_without_label_nb
```

	Text	Predicted
0	5000 ഉള്ള പോൾ ലോഗ്ഡ്വൻ ഇപ്പോള് 250000 എന്താ...	Fake
1	ഓഷോ രജനീഷ് പറഞ്ഞപോലെ എനിക്കപ്പോൾ തോന്നിയത് അ...	original
2	ചേട്ടാ വാർത്ത വയ്ക്കുന്നത് കേരളത്തിലാണ് സം...	Fake
3	Shame for entire Woman&#39	Fake
4	135 code janaghal andhu wide business cheythal...	Fake
...	...	...
1014	Correct ...China cheitha ...weapon spread ing ...	Fake
1015	ഈ WHO പറയുന്നതനുസരിച്ചു ചികിത്സയും ലോക്ക് ഡൌൺ ...	Fake
1016	Mask illa aarkum😋 😋	original
1017	ഇയാളെ കൊറോണ. രോഗി കൾ കിടയിൽ. ഇടാമായിരുന്നു---!!	Fake
1018	Kulathrikalkku badilayiAlavilathikal ...	original

1019 rows × 2 columns

```
results_without_label_nb.to_csv('output_file_task1.csv', index=False)
```