```
from google.colab import files
uploaded = files.upload()
```

Choose Files federalist.csv

- **federalist.csv**(text/csv) - 1100616 bytes, last modified: 11/12/2022 - 100% done

```
Saving federalist.csv to federalist.csv
```

```
import pandas as pd

df = pd.read_csv('federalist.csv', header=0, encoding='latin-1')
```

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
from nltk.corpus import stopwords

from sklearn.feature_extraction.text import TfidfVectorizer

stopwords = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(min_df=2, max_df=0.5, ngram_range=(1,1))
X = df.text
y = df.author
X.head()
map ={}

for i in df.author:
  if i in map:
    map[i] = map[i] + 1
  else:
    map[i] = 1

print(map)
```

```
{'HAMILTON': 49, 'JAY': 5, 'MADISON': 15, 'HAMILTON AND MADISON': 3, 'HAMILTON O
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0.

print(X_train.shape)
```

```
print(X_test.shape)
```

```
(66,)
(17,)
```

```
# apply tfidf vectorizer
X_train = vectorizer.fit_transform(X_train)  # fit and transform the train data
X_test = vectorizer.transform(X_test)        # transform only the test data
```

```
# take a peek at the data
# this is a very sparse matrix because most of the 8613 words don't occur in each sms

print('train size:', X_train.shape)
print(X_train.toarray()[:5])

print('\ntest size:', X_test.shape)
print(X_test.toarray()[:5])
```

```
train size: (66, 4603)
[[0.         0.03220702 0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.05589859 0.        ]]

test size: (17, 4603)
[[0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.02659749 0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]]
```

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import BernoulliNB

naive_bayes = BernoulliNB()
naive_bayes.fit(X_train, y_train)
```

```
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
```

```
prior_p = sum(y_train == 1)/len(y_train)
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, c

# make predictions on the test data
pred = naive_bayes.predict(X_test)

# print confusion matrix
```

```
print(confusion_matrix(y_test, pred))
```

```
[[10  0  0  0]
 [ 3  0  0  0]
 [ 2  0  0  0]
 [ 2  0  0  0]]
```

```
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
accuracy score:  0.5882352941176471
```

```
vectorizer = TfidfVectorizer(min_df=2, max_df=0.5, max_features=1000,ngram_range=(1,2)
X = df.text
y = df.author
X.head()
```

```
0    FEDERALIST. No. 1 General Introduction For the...
1    FEDERALIST No. 2 Concerning Dangers from Forei...
2    FEDERALIST No. 3 The Same Subject Continued (C...
3    FEDERALIST No. 4 The Same Subject Continued (C...
4    FEDERALIST No. 5 The Same Subject Continued (C...
Name: text, dtype: object
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0.

print(X_train.shape)
print(X_test.shape)
```

```
(66,)
(17,)
```

```
# apply tfidf vectorizer
X_train = vectorizer.fit_transform(X_train)  # fit and transform the train data
X_test = vectorizer.transform(X_test)        # transform only the test data
```

```
from sklearn.naive_bayes import BernoulliNB

naive_bayes = BernoulliNB()
naive_bayes.fit(X_train, y_train)
```

```
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
```

```
prior_p = sum(y_train == 1)/len(y_train)
```

```
pred = naive_bayes.predict(X_test)
```

```
# print confusion matrix
print(confusion_matrix(y_test, pred))
```

```
[[10  0  0  0]
 [ 0  3  0  0]
 [ 1  0  1  0]
 [ 0  0  0  2]]
```

```
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
accuracy score:  0.9411764705882353
```

```
!pip install "scikit_learn==0.22.2.post1"
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-whe
Requirement already satisfied: scikit_learn==0.22.2.post1 in /usr/local/lib/pyth
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.7/dist-pa
```

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, 

df = pd.read_csv('federalist.csv', header=0, encoding='latin-1')
# set up X and y
X = df.text
y = df.author

# divide into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0.

vectorizer = TfidfVectorizer(binary=True)
X_train = vectorizer.fit_transform(X_train)  # fit and transform the train data
X_test = vectorizer.transform(X_test)        # transform only the test data

#train
classifier = LogisticRegression(multi_class='multinomial',solver='lbfgs', class_weight
classifier.fit(X_train, y_train)

# evaluate
pred = classifier.predict(X_test)
```

```
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
accuracy score:  0.5882352941176471
```

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, 
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer

df = pd.read_csv('federalist.csv', header=0, encoding='latin-1')
# set up X and y
X = df.text
y = df.author

from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer

stopwords = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(stop_words=stopwords, binary=True)


X = vectorizer.fit_transform(df.text)
y = df.author

# divide into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0.
from sklearn.neural_network import MLPClassifier
classifier = MLPClassifier(solver='lbfgs', alpha=1e-5,
                   hidden_layer_sizes=(17, 2), random_state=1)
classifier.fit(X_train, y_train)
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score
pred = classifier.predict(X_test)
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
accuracy score:  0.5882352941176471
```

```
df = pd.read_csv('federalist.csv', header=0, encoding='latin-1')
# set up X and y
X = df.text
y = df.author

from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer

stopwords = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(stop_words=stopwords, binary=True)
```

```
X = vectorizer.fit_transform(df.text)
y = df.author

# divide into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0.
from sklearn.neural_network import MLPClassifier
classifier = MLPClassifier(solver='lbfgs', alpha=1e-5,
                   hidden_layer_sizes=(10, 2), random_state=1)
classifier.fit(X_train, y_train)
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score
pred = classifier.predict(X_test)
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
accuracy score:  0.5882352941176471
```

```
df = pd.read_csv('federalist.csv', header=0, encoding='latin-1')
# set up X and y
X = df.text
y = df.author

from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer

stopwords = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(stop_words=stopwords, binary=True)


X = vectorizer.fit_transform(df.text)
y = df.author

# divide into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0.
from sklearn.neural_network import MLPClassifier
classifier = MLPClassifier(solver='lbfgs', alpha=1e-5,
                   hidden_layer_sizes=(20, 4), random_state=1)
classifier.fit(X_train, y_train)
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score
pred = classifier.predict(X_test)
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
accuracy score:  0.17647058823529413
```

Colab paid products - Cancel contracts here

✓ 0s completed at 10:28 PM