```
# 1
import pandas as pd
from pandas.api.types import CategoricalDtype
import re
from nltk import word_tokenize

df = pd.read_csv('federalist.csv')
df.astype({'author': 'category'})
print(df[:])
```

```
          author                                               text
    0   HAMILTON  FEDERALIST. No. 1 General Introduction For the...
    1        JAY  FEDERALIST No. 2 Concerning Dangers from Forei...
    2        JAY  FEDERALIST No. 3 The Same Subject Continued (C...
    3        JAY  FEDERALIST No. 4 The Same Subject Continued (C...
    4        JAY  FEDERALIST No. 5 The Same Subject Continued (C...
    ..       ...                                                ...
    78  HAMILTON  FEDERALIST No. 79 The Judiciary Continued From...
    79  HAMILTON  FEDERALIST No. 80 The Powers of the Judiciary ...
    80  HAMILTON  FEDERALIST. No. 81 The Judiciary Continued, an...
    81  HAMILTON  FEDERALIST No. 82 The Judiciary Continued From...
    82  HAMILTON  FEDERALIST No. 83 The Judiciary Continued in R...

    [83 rows x 2 columns]
```

```
# 2
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
X = df['text']
y = df['author']
# display shape of train  & test
X_train.shape, y_train.shape
```

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

```
# 3 Bernoulli Naïve Bayes model
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk import word_tokenize
import nltk
nltk.download('stopwords')
nltk.download('punkt')

stopwords = set(stopwords.words("english"))

vectorizer = TfidfVectorizer(stop_words=stopwords)

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

```
# display shape of train  & test
print('train size:', X_train.shape)
print('\ntest size:', X_test.shape)
```

```
train size: (66, 7876)

test size: (17, 7876)
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
# 4
import math
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,

naive_bayes = MultinomialNB()
naive_bayes.fit(X_train, y_train)

prior_p = sum(y_train == 1)/len(y_train)
naive_bayes.class_log_prior_[1]
naive_bayes.feature_log_prob_


# make predictions on the test data
pred = naive_bayes.predict(X_test)
# print confusion matrix
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
accuracy score:  0.5882352941176471
```

```
# 5 - Bernoulli Naïve Bayes model
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk import word_tokenize
import nltk
import math
from sklearn.naive_bayes import BernoulliNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
nltk.download('stopwords')
nltk.download('punkt')

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0
stopwords = set(stopwords.words("english"))
vectorizer = TfidfVectorizer(stop_words=stopwords, max_features=1000, ngram_range=(1,
# apply tfidf vectorizer
```

```
X_train = vectorizer.fit_transform(X_train)  # fit and transform train
X_test = vectorizer.transform(X_test)        # transform only on test

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

# make predictions on the test data
pred = naive_bayes.predict(X_test)
# print confusion matrix
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
accuracy score:  0.9411764705882353
```

```
# 6 - Logistic Regression
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk import word_tokenize
import nltk
import math
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
nltk.download('stopwords')
nltk.download('punkt')

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0
stopwords = set(stopwords.words("english"))
vectorizer = TfidfVectorizer(stop_words=stopwords)

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

logisticReg = LogisticRegression(class_weight='balanced', random_state=0)
logisticReg.fit(X_train, y_train)

# make predictions on the test data
pred = logisticReg.predict(X_test)
# print confusion matrix
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
accuracy score:  0.7058823529411765
```

```
# 7- Neural Network
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk import word_tokenize
import nltk
import math
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
nltk.download('stopwords')
nltk.download('punkt')

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, train_size=0
stopwords = set(stopwords.words("english"))
vectorizer = TfidfVectorizer(stop_words=stopwords)

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

nuralNetwork = MLPClassifier(solver='lbfgs', random_state=3)
nuralNetwork.fit(X_train, y_train)

# make predictions on the test data
pred = nuralNetwork.predict(X_test)
# print confusion matrix
print('accuracy score: ', accuracy_score(y_test, pred))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
accuracy score:  0.8235294117647058
```

Colab paid products - Cancel contracts here

✓ 6s completed at 11:40 PM