Jesse Truong, JTT190006, 11/12/2022

```
import nltk
```

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

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Collecting scikit_learn==0.22.2.post1
  Downloading scikit_learn-0.22.2.post1-cp37-cp37m-manylinux1_x86_64.whl (7.1 MB)
     |████████████████████████████████| 7.1 MB 6.7 MB/s
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.7/dist-packag
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.0.2
    Uninstalling scikit-learn-1.0.2:
      Successfully uninstalled scikit-learn-1.0.2
ERROR: pip's dependency resolver does not currently take into account all the packag
yellowbrick 1.5 requires scikit-learn>=1.0.0, but you have scikit-learn 0.22.2.post1
imbalanced-learn 0.8.1 requires scikit-learn>=0.24, but you have scikit-learn 0.22.2
Successfully installed scikit-learn-0.22.2.post1
WARNING: The following packages were previously imported in this runtime:
  [sklearn]
You must restart the runtime in order to use newly installed versions.
```

RESTART RUNTIME

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

from sklearn.neural_network import MLPClassifier
from sklearn.pipeline import Pipeline

from sklearn.linear_model.logistic import LogisticRegression


df = pd.read_csv('federalist.csv')


df['author'] = df.author.astype('category')
print(df.head())
print()
print(df.groupby(['author']).count())
```

✓ 11s completed at 7:00 PM

```
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...
```

```
                      text
author
HAMILTON                49
HAMILTON AND MADISON     3
HAMILTON OR MADISON     11
JAY                      5
MADISON                 15
```

```
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['author'], test_size=0.
print(X_train.shape)
print(X_test.shape)
```

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

```
stoplist = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(stop_words=stoplist,)
X_train = vectorizer.fit_transform(X_train)  # fit and transform the train data
X_test = vectorizer.transform(X_test)        # transform only the test data
```

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

```
(66, 7876)
(17, 7876)
```

```
bnb = BernoulliNB()
bnb.fit(X_train, y_train) # Bernulli Navie Bayes model training
```

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

```
# make predictions on the test data
pred1 = bnb.predict(X_test)
print('accuracy score: ', accuracy_score(y_test, pred1))
```

```
accuracy score:  0.5882352941176471
```

```
tdf = TfidfVectorizer(stop_words=stoplist, max_features= 1000, ngram_range = (1,2)) # Remo
X2_train, X2_test, y2_train, y2_test = train_test_split(df['text'], df['author'], test_siz
X2_train = tdf.fit_transform(X2_train)
X2_test = tdf.transform(X2_test)

bnb = BernoulliNB()
bnb.fit(X2_train, y2_train) # Bernoulli Navies Bayes Model traning

pred2 = bnb.predict(X2_test)
print('accuracy score: ', accuracy_score(y2_test, pred2))
```

```
     accuracy score:  0.9411764705882353
```

```
print('accuracy score: ', accuracy_score(y_test, pred1))
print('accuracy score: ', accuracy_score(y2_test, pred2))
```

```
     accuracy score:  0.5882352941176471
     accuracy score:  0.9411764705882353
```

```
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['author'], test_size=0.
```

```
pipe1 = Pipeline([
        ('tfidf', TfidfVectorizer()),
        ('logreg', LogisticRegression(multi_class='multinomial',solver='saga',class_weight
]) # Pipeline for logistic Regression with multi classing
pipe1.fit(X_train,y_train) # Logistic Regression Training
pred3 = pipe1.predict(X_test)
import numpy as np
print("\nOverall accuracy: ", np.mean(pred3==y_test))
```

```
     Overall accuracy:  1.0
     /usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_sag.py:330: ConvergenceI
       "the coef_ did not converge", ConvergenceWarning)
```

```
pipe2 = Pipeline([
        ('tfidf', TfidfVectorizer()),
        ('neuralnet', MLPClassifier(solver='lbfgs', alpha=1e-5,
                    hidden_layer_sizes=(50, 10), random_state=1234)),
         ]) # Pipeline for Neural Network

pipe2.fit(X_train, y_train) # NN training
pred4 = pipe2.predict(X_test)
print("\nOverall accuracy: ", np.mean(pred4==y_test))
```

```
     Overall accuracy:  0.8823529411764706
```

Colab paid products - Cancel contracts here