

▼ Import statements

```
#@title Import statements
import pandas as pd
import nltk
from collections import Counter
from sklearn.model_selection import train_test_split
from nltk.corpus import stopwords
from sklearn.naive_bayes import BernoulliNB
from sklearn import metrics
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
nltk.download('stopwords')

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

▼ Reading in federalist.csv

```
#@title Reading in federalist.csv
df = pd.read_csv('federalist.csv')
```

▼ Create categorical label for each author

```
#@title Create categorical label for each author
authors = df['author'].unique().tolist()
labels = {}
for author in authors:
    labels[author] = authors.index(author) + 1

df['label'] = [labels[author] for author in df['author'].tolist()]
```

Resources

You are not subscribed.
[Learn more.](#)

You currently have zero compute units available. Resources offered free of charge are not guaranteed. Purchase more units [here](#).

[Manage sessions](#)

Want more memory
and disk space?

[Upgrade to Colab Pro](#)

Python 3 Google
Compute Engine
backend

Showing resources
since 4:41 PM

System RAM



Disk



```
# Print the first few rows of the dataframe
print(df.head())
```

	author	text	label
0	HAMILTON	FEDERALIST. No. 1 General Introduction For the...	1
1	JAY	FEDERALIST No. 2 Concerning Dangers from Forei...	2
2	JAY	FEDERALIST No. 3 The Same Subject Continued (C...	2
3	JAY	FEDERALIST No. 4 The Same Subject Continued (C...	2
4	JAY	FEDERALIST No. 5 The Same Subject Continued (C...	2

▼ Printing counts for each author

```
#@title Printing counts for each author
counts_by_author = Counter(name for name in df['author'].tolist())
for author, freq in counts_by_author.items():
    print(f'{author}: {freq}')
```

```
HAMILTON: 49
JAY: 5
MADISON: 15
HAMILTON AND MADISON: 3
HAMILTON OR MADISON: 11
```

▼ Unique stopwords and vectorizer initialization

```
#@title Unique stopwords and vectorizer initialization
stopword = set(stopwords.words('english'))
vectorizer = TfidfVectorizer(stop_words=stopword)
```

```
X = df.text
Y = df.label
```

▼ Creating 80-20 train, test split using 1234 as random state

```
#@title Creating 80-20 train, test split using 1234 as random state
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, train_size=0.8, random_state=1234)
```

▼ Printing the shape of the train and test

```
#@title Printing the shape of the train and test
print(X_train.shape)
print(Y_train.shape)

(66,)
(66,)
```

```
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
```

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

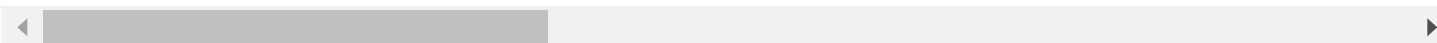
```
BernoulliNB()
```

▼ Prediction using Naive Bayes

```
#@title Prediction using Naive Bayes
pred = naive_bayes.predict(X_test)
print(metrics.classification_report(Y_test, pred))
print()
print("Overall accuracy: ", accuracy_score(Y_test, pred))
```

	precision	recall	f1-score	support
1	0.59	1.00	0.74	10
2	0.00	0.00	0.00	2
3	0.00	0.00	0.00	2
5	0.00	0.00	0.00	3
accuracy			0.59	17
macro avg	0.15	0.25	0.19	17
weighted avg	0.35	0.59	0.44	17

```
Overall accuracy: 0.5882352941176471
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision is ill-defined for classes in labels [1] that have no predicted samples. The precision value for these classes has been set to nan.
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision is ill-defined for classes in labels [1] that have no predicted samples. The precision value for these classes has been set to nan.
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision is ill-defined for classes in labels [1] that have no predicted samples. The precision value for these classes has been set to nan.
_warn_prf(average, modifier, msg_start, len(result))
```



▼ Updating vectorizer

```
#@title Updating vectorizer
vectorizer = TfidfVectorizer(stop_words=stopword, max_features=1000, ngram_range=(1,2))
X_train, X_test, Y_train, Y_test = train_test_split(df['text'], df['label'], train_size=0.8, random_state=42)
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
naive_bayes = BernoulliNB()
naive_bayes.fit(X_train, Y_train)
pred = naive_bayes.predict(X_test)
print(metrics.classification_report(Y_test, pred))
print()
print("Overall accuracy: ", accuracy_score(Y_test, pred))
```

	precision	recall	f1-score	support
1	0.91	1.00	0.95	10
2	1.00	0.50	0.67	2
3	1.00	1.00	1.00	2
5	1.00	1.00	1.00	3
accuracy			0.94	17
macro avg	0.98	0.88	0.90	17
weighted avg	0.95	0.94	0.93	17

```
Overall accuracy: 0.9411764705882353
```

▼ Logistic Regression (no parameters)

```
#@title Logistic Regression (no parameters)
Logistic_regression = LogisticRegression(multi_class='multinomial', solver='lbfgs')
Logistic_regression.fit(X_train, Y_train)
pred = Logistic_regression.predict(X_test)
print(metrics.classification_report(Y_test, pred))
print()
print("Accuracy: ", accuracy_score(Y_test, pred))
```

	precision	recall	f1-score	support
1	0.59	1.00	0.74	10
2	0.00	0.00	0.00	2
3	0.00	0.00	0.00	2
5	0.00	0.00	0.00	3
accuracy			0.59	17
macro avg	0.15	0.25	0.19	17
weighted avg	0.35	0.59	0.44	17

```
Accuracy: 0.5882352941176471
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision is zero.
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision is zero.
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision is zero.
_warn_prf(average, modifier, msg_start, len(result))
```

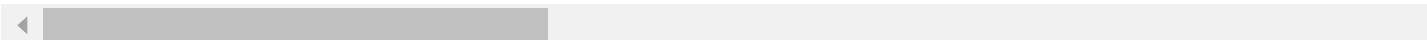
▼ Logistic Regression (changed solver to saga, and class_weight=balanced)

```
#@title Logistic Regression (changed solver to saga, and class_weight=balanced)
changed_logistic_regression = LogisticRegression(multi_class='multinomial', solver='saga', class_weight='balanced')
changed_logistic_regression.fit(X_train, Y_train)
pred = changed_logistic_regression.predict(X_test)
print(metrics.classification_report(Y_test, pred))
```

```
print()
print("Accuracy: ", accuracy_score(Y_test, pred))
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	10
2	1.00	1.00	1.00	2
3	0.00	0.00	0.00	2
4	0.00	0.00	0.00	0
5	0.75	1.00	0.86	3
accuracy			0.88	17
macro avg	0.55	0.60	0.57	17
weighted avg	0.84	0.88	0.86	17

```
Accuracy: 0.8823529411764706
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_sag.py:354: ConvergenceWarning: T
ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))
```



▼ Using neural network

```
#@title Using neural network
from sklearn.neural_network import MLPClassifier
nn = MLPClassifier(solver='lbfgs', alpha=2e-5, hidden_layer_sizes=(20, 15), random_state=1234)
nn.fit(X_train, Y_train)
pred = nn.predict(X_test)
```

```
print(metrics.classification_report(Y_test, pred))
print()
print("Accuracy: ", accuracy_score(Y_test, pred))
```

	precision	recall	f1-score	support
1	0.77	1.00	0.87	10
2	0.00	0.00	0.00	2
3	0.50	0.50	0.50	2
5	1.00	0.67	0.80	3
accuracy			0.76	17
macro avg	0.57	0.54	0.54	17
weighted avg	0.69	0.76	0.71	17

Accuracy: 0.7647058823529411

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision is zero for some classes.
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Recall is zero for some classes.
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: F-score is zero for some classes.
_warn_prf(average, modifier, msg_start, len(result))
```

▼ Trying different topologies with NN

```
#@title Trying different topologies with NN
# Using different activation, solver, learning rate and hidden_layer sizes
nnu = MLPClassifier(activation='tanh', solver='adam', alpha=1e-6, hidden_layer_sizes=(100, 20), random_state=42)
nnu.fit(X_train, Y_train)
pred = nnu.predict(X_test)
print(metrics.classification_report(Y_test, pred))
print()
print("Accuracy: ", accuracy_score(Y_test, pred))
```

	precision	recall	f1-score	support
1	0.77	1.00	0.87	10

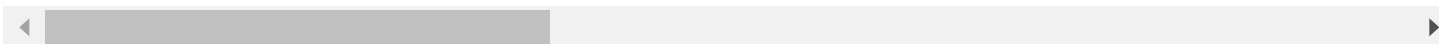
2	0.00	0.00	0.00	2
3	1.00	0.50	0.67	2
5	1.00	1.00	1.00	3
accuracy			0.82	17
macro avg	0.69	0.62	0.63	17
weighted avg	0.75	0.82	0.77	17

Accuracy: 0.8235294117647058

```

/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:696: C
ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetri
_warn_prf(average, modifier, msg_start, len(result))

```



[Change runtime type](#)

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 4:42 PM

