requirements:

numpy, pandas, nltk, scikit-learn, matplotlib, seaborn

In [1]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.datasets import fetch_20newsgroups
from nltk.corpus import names
from nltk.stem import WordNetLemmatizer
```

In [2]:

```
# Load Data
all_names = set(names.words())
lemmatizer = WordNetLemmatizer()

def letters_only(astr):
    for c in astr:
        if not c.isalpha():
            return False
    return True

def clean_text(docs):
    cleaned_docs = []
    for doc in docs:
        cleaned_docs.append(' '.join([lemmatizer.lemmatize(word.lower())
                                      for word in doc.split()
                                      if letters_only(word)
                                      and word not in all_names]))
    return cleaned_docs
```

In [3]:

```python
# Binary classification
categories = ['comp.graphics', 'sci.space']

data_train = fetch_20newsgroups(subset='train', categories=categories, random_state=42)
data_test = fetch_20newsgroups(subset='test', categories=categories, random_state=42)

cleaned_train = clean_text(data_train.data)
label_train = data_train.target
cleaned_test = clean_text(data_test.data)
label_test = data_test.target

from collections import Counter
Counter(label_train)

tfidf_vectorizer = TfidfVectorizer(sublinear_tf=True, max_df=0.5, stop_words='english', max_features=8000)
term_docs_train = tfidf_vectorizer.fit_transform(cleaned_train)
term_docs_test = tfidf_vectorizer.transform(cleaned_test)

from sklearn.svm import SVC
svm = SVC(kernel='linear', C=1.0, random_state=42)
svm.fit(term_docs_train, label_train)
accuracy = svm.score(term_docs_test, label_test)
print('The accuracy on testing set is: {0:.1f}%'.format(accuracy*100))
```

```
The accuracy on testing set is: 96.4%
```

In [4]:

```
# Multiclass classification
categories = [
    'alt.atheism',
    'talk.religion.misc',
    'comp.graphics',
    'sci.space',
    'rec.sport.hockey'
]
data_train = fetch_20newsgroups(subset='train', categories=categories, random_
state=42)
data_test = fetch_20newsgroups(subset='test', categories=categories, random_st
ate=42)

cleaned_train = clean_text(data_train.data)
label_train = data_train.target
cleaned_test = clean_text(data_test.data)
label_test = data_test.target

term_docs_train = tfidf_vectorizer.fit_transform(cleaned_train)
term_docs_test = tfidf_vectorizer.transform(cleaned_test)

svm = SVC(kernel='linear', C=1.0, random_state=42)
svm.fit(term_docs_train, label_train)
accuracy = svm.score(term_docs_test, label_test)
print('The accuracy on testing set is: {0:.1f}%'.format(accuracy*100))

from sklearn.metrics import classification_report
prediction = svm.predict(term_docs_test)
report = classification_report(label_test, prediction)
print(report)
```

```
The accuracy on testing set is: 88.6%
              precision    recall  f1-score   support

           0       0.81      0.77      0.79       319
           1       0.91      0.94      0.93       389
           2       0.98      0.96      0.97       399
           3       0.93      0.93      0.93       394
           4       0.73      0.76      0.74       251

   micro avg       0.89      0.89      0.89      1752
   macro avg       0.87      0.87      0.87      1752
weighted avg       0.89      0.89      0.89      1752
```

In [5]:

```
# Grid search

categories = None
data_train = fetch_20newsgroups(subset='train', categories=categories, random_
state=42)
data_test = fetch_20newsgroups(subset='test', categories=categories, random_st
ate=42)

cleaned_train = clean_text(data_train.data)
label_train = data_train.target
cleaned_test = clean_text(data_test.data)
label_test = data_test.target

tfidf_vectorizer = TfidfVectorizer(sublinear_tf=True, max_df=0.5, stop_words=
'english', max_features=8000)
term_docs_train = tfidf_vectorizer.fit_transform(cleaned_train)
term_docs_test = tfidf_vectorizer.transform(cleaned_test)

parameters = {'C': [0.1, 1, 10, 100]}
svc_libsvm = SVC(kernel='linear')

from sklearn.model_selection import GridSearchCV
grid_search = GridSearchCV(svc_libsvm, parameters, n_jobs=-1, cv=3)


import timeit
start_time = timeit.default_timer()
grid_search.fit(term_docs_train, label_train)
print("--- %0.3fs seconds ---" % (timeit.default_timer() - start_time))

print(grid_search.best_params_)
print(grid_search.best_score_)

svc_libsvm_best = grid_search.best_estimator_
accuracy = svc_libsvm_best.score(term_docs_test, label_test)
print('The accuracy on testing set is: {0:.1f}%'.format(accuracy*100))


from sklearn.svm import LinearSVC
svc_linear = LinearSVC()
grid_search = GridSearchCV(svc_linear, parameters, n_jobs=-1, cv=3)

start_time = timeit.default_timer()
grid_search.fit(term_docs_train, label_train)
print("--- %0.3fs seconds ---" % (timeit.default_timer() - start_time))

print(grid_search.best_params_)
print(grid_search.best_score_)
svc_linear_best = grid_search.best_estimator_
accuracy = svc_linear_best.score(term_docs_test, label_test)
print('The accuracy on testing set is: {0:.1f}%'.format(accuracy*100))
```

```
--- 370.753s seconds ---
{'C': 10}
0.8665370337634789
The accuracy on testing set is: 76.2%
--- 10.242s seconds ---
{'C': 1}
0.8707795651405339
The accuracy on testing set is: 77.9%
```

In [6]:

```
# Pipeline
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('tfidf', TfidfVectorizer(stop_words='english')),
    ('svc', LinearSVC()),
])

parameters_pipeline = {
    'tfidf__max_df': (0.25, 0.5),
    'tfidf__max_features': (40000, 50000),
    'tfidf__sublinear_tf': (True, False),
    'tfidf__smooth_idf': (True, False),
    'svc__C': (0.1, 1, 10, 100),
}

grid_search = GridSearchCV(pipeline, parameters_pipeline, n_jobs=-1, cv=3)

start_time = timeit.default_timer()
grid_search.fit(cleaned_train, label_train)
print("--- %0.3fs seconds ---" % (timeit.default_timer() - start_time))

print(grid_search.best_params_)
print(grid_search.best_score_)
pipeline_best = grid_search.best_estimator_
accuracy = pipeline_best.score(cleaned_test, label_test)
print('The accuracy on testing set is: {0:.1f}%'.format(accuracy*100))
```

```
--- 496.068s seconds ---
{'svc__C': 1, 'tfidf__max_df': 0.5, 'tfidf__max_features': 40000, 'tfidf__smo
oth_idf': False, 'tfidf__sublinear_tf': True}
0.8883683931412409
The accuracy on testing set is: 80.6%
```

In [ ]: