```
from nltk.stem import WordNetLemmatizer
from nltk import word_tokenize
import numpy as np
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import TruncatedSVD
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import Normalizer
import re
```

```
from sklearn.datasets import fetch_20newsgroups
full_ds = fetch_20newsgroups(subset='all')
print (len(full_ds.target))
```

```
18846
```

В TfidfVectorizer не зашита лемматизация, вопользуемся инструментом из nltk, переопределим лемматизатор и используем его при преобразовании в вектор.

```
class LemmaTokenizer(object):
    def __init__(self):
        self.wnl = WordNetLemmatizer()
    def __call__(self, doc):
        return [self.wnl.lemmatize(t) for t in word_tokenize(doc)]
```

Также отфильтруем стоп-слова из словаря (словарь найден на просторах интернета).

```
stop_words = open('data/stop_words.txt', 'r').readlines()
len(stop_words)
```

```
544
```

```
full_ds.target_names[0:4]
```

```
['alt.atheism',
 'comp.graphics',
 'comp.os.ms-windows.misc',
 'comp.sys.ibm.pc.hardware']
```

```
ds = fetch_20newsgroups(subset='all', categories = full_ds.target_names[0:4], remove=
('headers', 'footers', 'quotes'))
```

```
data_prep = []
for el in ds.data:
    filtered = re.findall(u'(?u)\\b\\w\\w+\\b', el)
    el_prep = ' '.join(filtered)
    data_prep.append(el_prep)
```

Tfidf с лемматизацией и сингулярное разложение.

```
stopwords = []
for el in stop_words:
    stopwords.append(el.strip('\n'))
```

```
tfidf_vectorizer = TfidfVectorizer(stop_words=stopwords, tokenizer=LemmaTokenizer())
tfidf_data = tfidf_vectorizer.fit_transform(data_prep)
```

```
topics = 4
svd = TruncatedSVD(topics)
normalizer = Normalizer(copy=False)
lsa = make_pipeline(svd, normalizer)
X = lsa.fit_transform(tfidf_data)
explained_variance = svd.explained_variance_ratio_.sum()
```

```
print(explained_variance)
```

```
0.0231372167648
```

Разобьем данные на тренировочную и тестовую выборки.

```
from sklearn.cross_validation import train_test_split
y = ds.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_sta
te = 1)

N_train, _ = X_train.shape
N_test,  _ = X_test.shape

print (N_train, N_test)
```

```
(2617, 1122)
```

Обучим классификатор.

```
knn = KNeighborsClassifier().fit(X_train, y_train)

y_train_predict = knn.predict(X_train)
y_test_predict = knn.predict(X_test)
```

Оценим ошибку.

```
err_train = np.mean(y_train != y_train_predict)
err_test  = np.mean(y_test  != y_test_predict)
print (err_train, err_test)
print (classification_report(y_test, y_test_predict))
```

```
(0.23920519679021782, 0.31194295900178254)
             precision    recall  f1-score   support

          0       0.81      0.92      0.86       246
          1       0.57      0.63      0.60       278
          2       0.61      0.53      0.56       285
          3       0.76      0.71      0.73       313

avg / total       0.69      0.69      0.68      1122
```

```
for i, topic in enumerate(svd.components_, start=1):
    ind = np.argsort(topic)
    print 'The most popular words in hidden topic number {} are: {}'.format(i, np.asa
rray(tfidf_vectorizer.get_feature_names())[ind[-15:-1]][:])
```

```
The most popular words in hidden topic number 1 are: [u'scsi' u'work' u'don' u'doe'
 u'program' u'ha' u'system' u'driver'
 u'problem' u'wa' u'card' u'drive' u'a' u'file']
The most popular words in hidden topic number 2 are: [u'seagate' u'boot' u'hd' u'jump
er' u'bios' u'floppy' u'isa' u'card'
 u'hard' u'bus' u'disk' u'controller' u'ide' u'scsi']
The most popular words in hidden topic number 3 are: [u'claim' u'evidence' u'bible'
 u'moral' u'point' u'argument' u'atheism'
 u'don' u'belief' u'religion' u'people' u'atheist' u'wa' u'a']
The most popular words in hidden topic number 4 are: [u'tiff' u'floppy' u'directory'
 u'bmp' u'convert' u'hard' u'ide' u'gif'
 u'scsi' u'image' u'program' u'disk' u'format' u'drive']
```