Hashing_vectorizer

March 24, 2019

In [10]:
```
import pandas as pd
from sklearn.linear_model import SGDClassifier
from sklearn.feature_extraction.text import HashingVectorizer
from sklearn.datasets import fetch_20newsgroups
from scipy.sparse import csr_matrix
news_groups_dataset = fetch_20newsgroups(shuffle=True, remove=('headers','footers','qu
def streaming():
    for response, item in zip(news_groups_dataset.target, news_groups_dataset.data):
        yield response, item
hashing_trick = HashingVectorizer(stop_words='english', norm='l2', non_negative=True )
learner = SGDClassifier(random_state=101)
texts = list()
targets = list()
for n,(target, text) in enumerate(streaming()):
    texts.append(text)
    targets.append(target)
    if n % 1000 == 0 and n > 0:
        learning_chunk = hashing_trick.transform(texts)
        learner.partial_fit(learning_chunk, targets, classes=[k for k in range(20)])
        if n > 1000:
            last_validation_score = learner.score(learning_chunk, targets)
        texts, targets = list(), list()
print ("Last validation result : %f" % last_validation_score)
```

```
Last validation result : 0.949000
```

In [21]:
```
new_text = ['A 2014 red Toyota Prius v Five with fewer than 14k miles. Powered by\
            a reliable 1.8L four cylinder hybrid engin taht averges 44mpg\
            in the city and 40mpg on the higway.']
text_vector = hashing_trick.transform(new_text)
index = learner.predict(text_vector)
print("Predicted discussion group %s" % news_groups_dataset.target_names[int(index)])
```

```
Predicted discussion group rec.autos
```

In [ ]: