In [1]:
```
#task 2 simple text classification
def gender_features(word):
    return {'last_letter':word[-1]}
```

In [2]:
```
gender_features('Sharmila k')
```

Out[2]:
```
{'last_letter': 'k'}
```

In [3]:
```
from nltk.corpus import names
#names.words()
print(len(names.words()))
```

```
7944
```

In [4]:
```
labeled_names=([(name,'male')for name in names.words('male.txt')]+[(name,'female') f
```

In [5]:
```
import random
random.shuffle(labeled_names)
```

In [6]:
```
featuresets=[(gender_features(n),gender) for (n,gender) in labeled_names]
```

In [7]:
```
train_set,test_set=featuresets[:5000],featuresets[5000:]
```

In [8]:
```
import nltk
classifier=nltk.NaiveBayesClassifier.train(train_set)
```

In [10]:
```
classifier.classify(gender_features('Sharmila'))
```

Out[10]:
```
'female'
```

In [12]:
```
classifier.classify(gender_features('David'))
```

Out[12]:
```
'male'
```

In [13]:
```
print(nltk.classify.accuracy(classifier,test_set))
```

```
0.7666440217391305
```

In [14]:
```
#task 3 count vectorizer
from sklearn.feature_extraction.text import CountVectorizer
vect=CountVectorizer(binary=True)
```

In [15]:
```
corpus=["Tessaract is good optical character recognition engine  ", "optical charact
```

In [16]:
```
vect.fit(corpus)
```

Out[16]:
```
CountVectorizer(binary=True)
```

In [17]:
```
vocab=vect.vocabulary_
```

In [18]:
```
for key in sorted(vocab.keys()):
    print("{}:{}".format(key,vocab[key]))
```

```
character:0
engine:1
good:2
is:3
optical:4
recognition:5
significant:6
tessaract:7
```

In [19]:
```
print(vect.transform(["This is a good optical illusion"]).toarray())
```

```
[[0 0 1 1 1 0 0 0]]
```

In [0]:
```

```