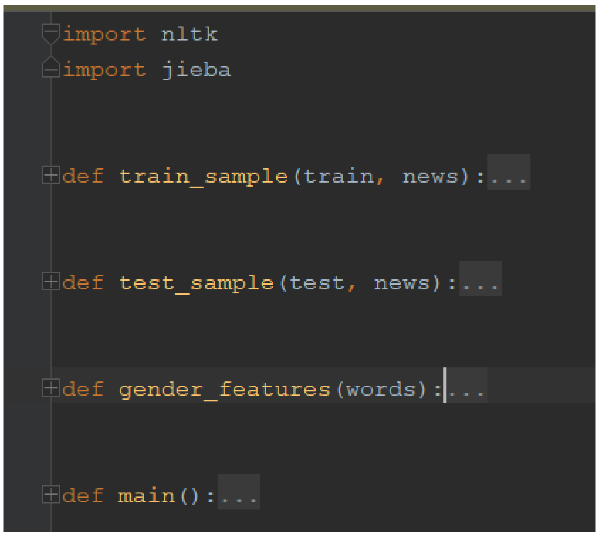
Stock market prediction report

Four main functions and two package as follow :



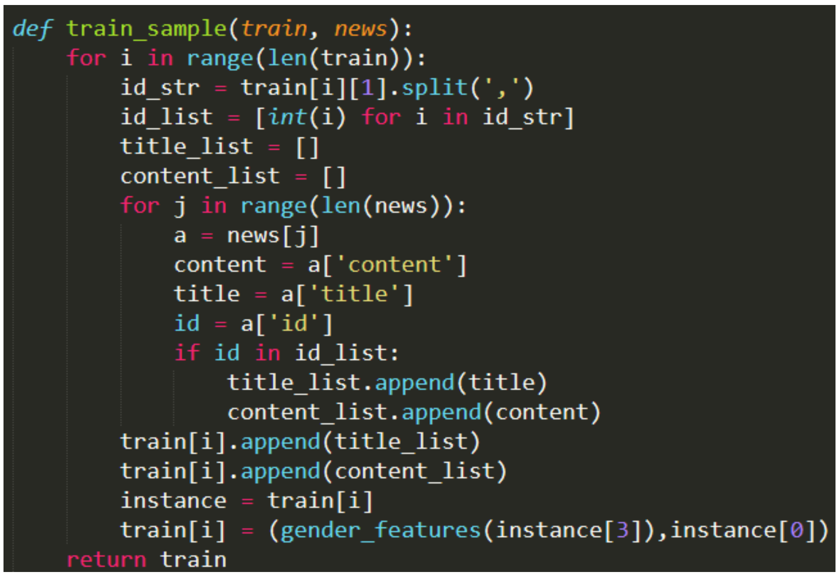
```
import nltk
import jieba


def train_sample(train, news):...


def test_sample(test, news):...


def gender_features(words):...


def main():...
```

Train_sample:

Input : the string information read from train.txt and news.txt

Do : find every specific news related to each company, and then generate the feature for classifier(use title or content)

return: list type, the data already for train which contain word features and labels.



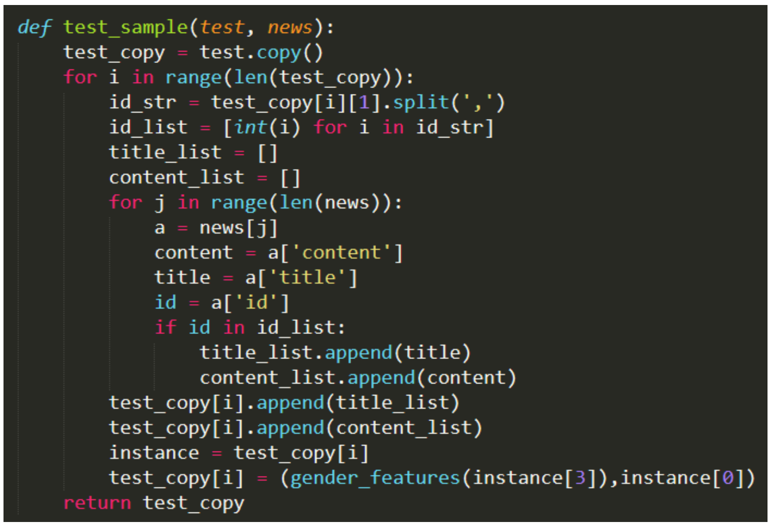
```
def train_sample(train, news):
    for i in range(len(train)):
        id_str = train[i][1].split(',')
        id_list = [int(i) for i in id_str]
        title_list = []
        content_list = []
        for j in range(len(news)):
            a = news[j]
            content = a['content']
            title = a['title']
            id = a['id']
            if id in id_list:
                title_list.append(title)
                content_list.append(content)
        train[i].append(title_list)
        train[i].append(content_list)
        instance = train[i]
        train[i] = (gender_features(instance[3]),instance[0])
    return train
```

test_sample:

Input : the string information read from test.txt and news.txt

Do : find every specific news related to each company by id, and then generate the feature for classifier(use title or content)

return: list type, the data already for test which contain word features but no labels.



```python
def test_sample(test, news):
    test_copy = test.copy()
    for i in range(len(test_copy)):
        id_str = test_copy[i][1].split(',')
        id_list = [int(i) for i in id_str]
        title_list = []
        content_list = []
        for j in range(len(news)):
            a = news[j]
            content = a['content']
            title = a['title']
            id = a['id']
            if id in id_list:
                title_list.append(title)
                content_list.append(content)
        test_copy[i].append(title_list)
        test_copy[i].append(content_list)
        instance = test_copy[i]
        test_copy[i] = (gender_features(instance[3]),instance[0])
    return test_copy
```

Gender_features:

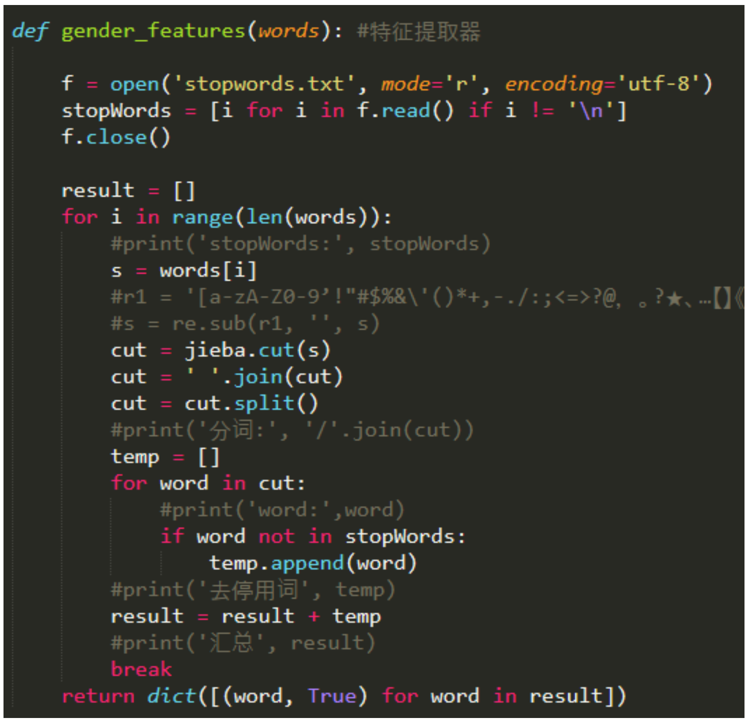
Input : a list of every news about one company

Do :

1.use jieba package to split sentences into several words

2.download a stopword list from internet, delete stopwords in previous step

return: a dict which structure is suitable for classifier



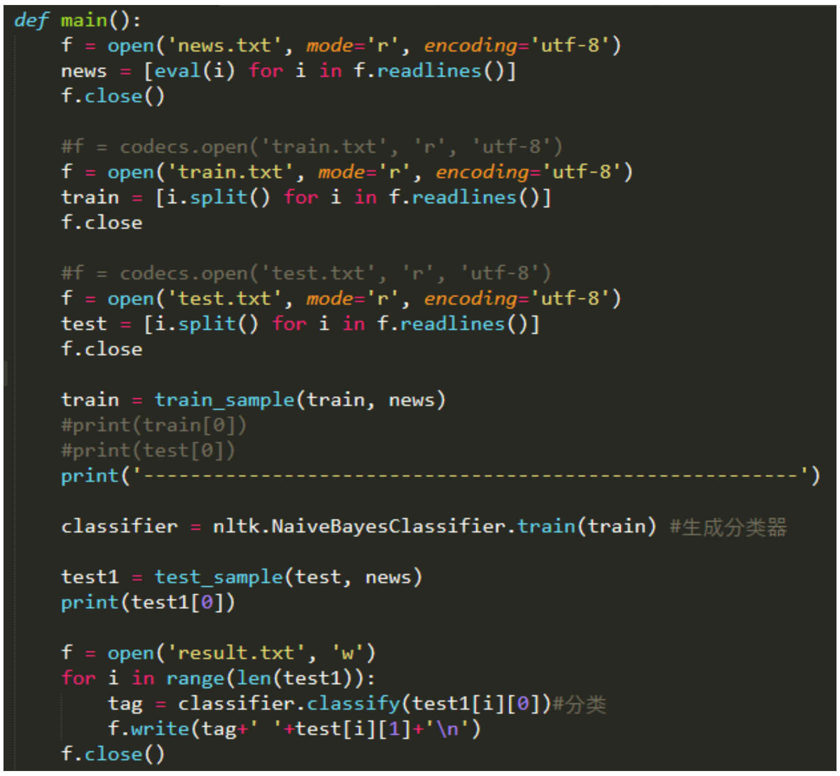
```python
def gender_features(words): #特征提取器

    f = open('stopwords.txt', mode='r', encoding='utf-8')
    stopWords = [i for i in f.read() if i != '\n']
    f.close()

    result = []
    for i in range(len(words)):
        #print('stopWords:', stopWords)
        s = words[i]
        #r1 = '[a-zA-Z0-9’!"#$%&\'()*+,-./:;<=>?@, 。?★、…【】《
        #s = re.sub(r1, '', s)
        cut = jieba.cut(s)
        cut = ' '.join(cut)
        cut = cut.split()
        #print('分词:', '/'.join(cut))
        temp = []
        for word in cut:
            #print('word:',word)
            if word not in stopWords:
                temp.append(word)
        #print('去停用词', temp)
        result = result + temp
        #print('汇总', result)
        break
    return dict([(word, True) for word in result])
```

main:

input: none

do:

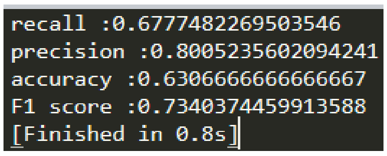
1. read news.txt, train.txt, test.txt
2. call the train_sample function to generate training data
3.use training data to train the naivebayes classifier
4.call the test_sample function to generate test data
5.feed the trained classifier with test data and get prediction, and wirte into result.txt at same time



```python
def main():
    f = open('news.txt', mode='r', encoding='utf-8')
    news = [eval(i) for i in f.readlines()]
    f.close()

    #f = codecs.open('train.txt', 'r', 'utf-8')
    f = open('train.txt', mode='r', encoding='utf-8')
    train = [i.split() for i in f.readlines()]
    f.close

    #f = codecs.open('test.txt', 'r', 'utf-8')
    f = open('test.txt', mode='r', encoding='utf-8')
    test = [i.split() for i in f.readlines()]
    f.close

    train = train_sample(train, news)
    #print(train[0])
    #print(test[0])
    print('---------------------------------------------------------')

    classifier = nltk.NaiveBayesClassifier.train(train) #生成分类器

    test1 = test_sample(test, news)
    print(test1[0])

    f = open('result.txt', 'w')
    for i in range(len(test1)):
        tag = classifier.classify(test1[i][0])#分类
        f.write(tag+' '+test[i][1]+'\n')
    f.close()
```

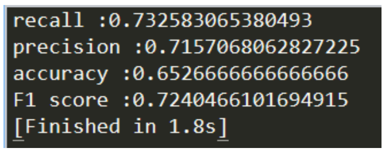
Result:

(use title):



```
recall :0.6777482269503546
precision :0.8005235602094241
accuracy :0.6306666666666667
F1 score :0.7340374459913588
[Finished in 0.8s]
```

(use content):



```
recall :0.732583065380493
precision :0.7157068062827225
accuracy :0.6526666666666666
F1 score :0.7240466101694915
[Finished in 1.8s]
```