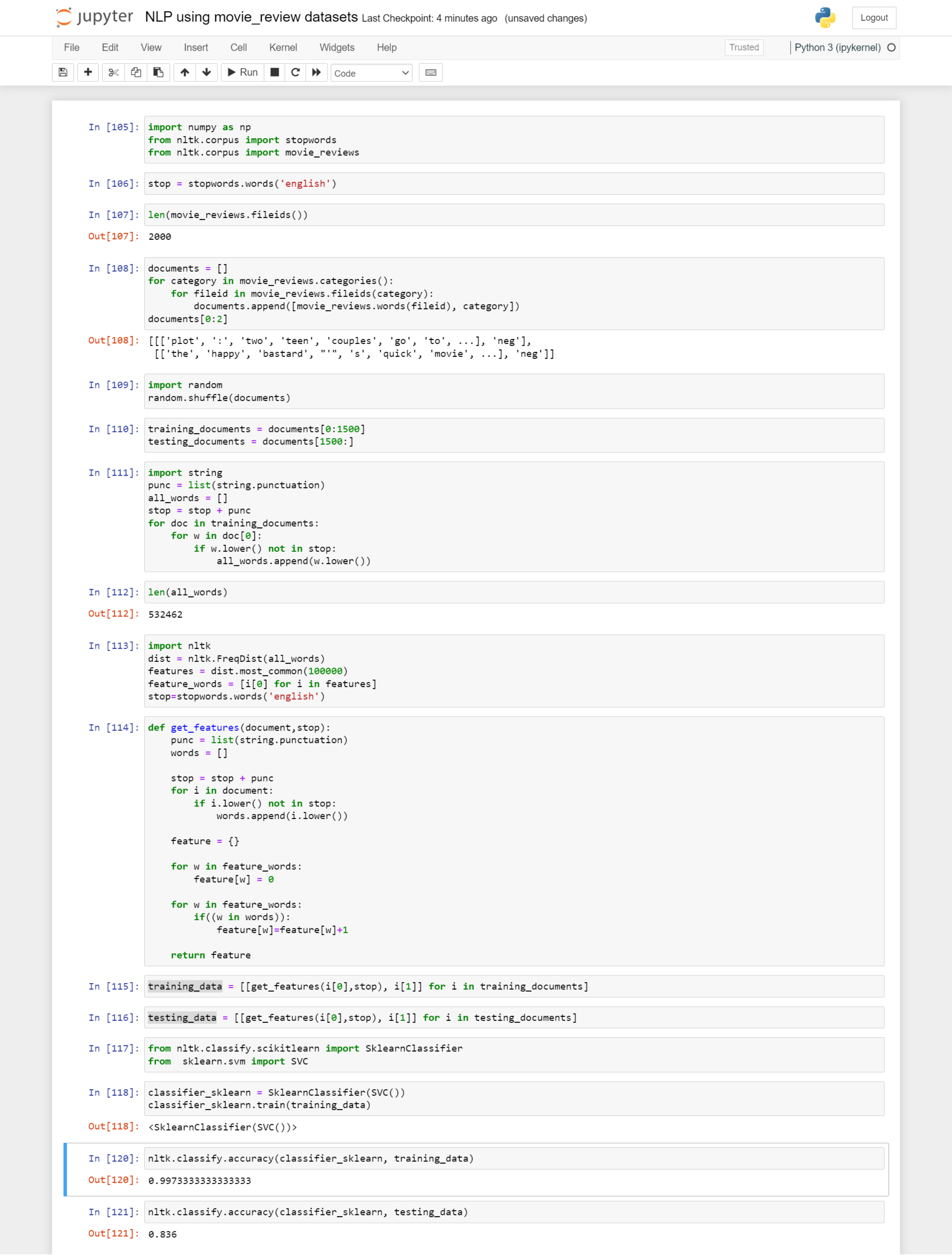

NLP using movie_review datasets Last Checkpoint: 4 minutes ago (unsaved changes)

```
In [105]: import numpy as np
          from nltk.corpus import stopwords
          from nltk.corpus import movie_reviews

In [106]: stop = stopwords.words('english')

In [107]: len(movie_reviews.fileids())

Out[107]: 2000

In [108]: documents = []
          for category in movie_reviews.categories():
              for fileid in movie_reviews.fileids(category):
                  documents.append([movie_reviews.words(fileid), category])
          documents[0:2]

Out[108]: [[['plot', ':', 'two', 'teen', 'couples', 'go', 'to', ...], 'neg'],
           [['the', 'happy', 'bastard', "'", 's', 'quick', 'movie', ...], 'neg']]

In [109]: import random
          random.shuffle(documents)

In [110]: training_documents = documents[0:1500]
          testing_documents = documents[1500:]

In [111]: import string
          punc = list(string.punctuation)
          all_words = []
          stop = stop + punc
          for doc in training_documents:
              for w in doc[0]:
                  if w.lower() not in stop:
                      all_words.append(w.lower())

In [112]: len(all_words)

Out[112]: 532462

In [113]: import nltk
          dist = nltk.FreqDist(all_words)
          features = dist.most_common(100000)
          feature_words = [i[0] for i in features]
          stop=stopwords.words('english')

In [114]: def get_features(document,stop):
              punc = list(string.punctuation)
              words = []

              stop = stop + punc
              for i in document:
                  if i.lower() not in stop:
                      words.append(i.lower())

              feature = {}

              for w in feature_words:
                  feature[w] = 0

              for w in feature_words:
                  if((w in words)):
                      feature[w]=feature[w]+1

              return feature

In [115]: training_data = [[get_features(i[0],stop), i[1]] for i in training_documents]

In [116]: testing_data = [[get_features(i[0],stop), i[1]] for i in testing_documents]

In [117]: from nltk.classify.scikitlearn import SklearnClassifier
          from  sklearn.svm import SVC

In [118]: classifier_sklearn = SklearnClassifier(SVC())
          classifier_sklearn.train(training_data)

Out[118]: <SklearnClassifier(SVC())>

In [120]: nltk.classify.accuracy(classifier_sklearn, training_data)

Out[120]: 0.9973333333333333

In [121]: nltk.classify.accuracy(classifier_sklearn, testing_data)

Out[121]: 0.836
```