```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import nltk

#----------------------------------------- Exploring names "corpus" (dataset, really)

>>> from nltk.corpus import names
>>> names.fileids()
['female.txt', 'male.txt']
>>> dir(names)
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getstate__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__',
'__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_citation', '_encoding',
'_fileids', '_get_root', '_license', '_readme', '_root', '_tagset', '_unload', 'abspath', 'abspaths',
'citation', 'encoding', 'ensure_loaded', 'fileids', 'license', 'open', 'raw', 'readme', 'root',
'words']
>>> names.raw('female.txt')[:100]
'Abagael\nAbagail\nAbbe\nAbbey\nAbbi\nAbbie\nAbby\nAbigael\nAbigail\nAbigale\nAbra\nAcacia\nAda\nAdah
\nAdaline\nAdar'
>>> print(names.raw('female.txt')[:100])
Abagael
Abagail
Abbe
Abbey
Abbi
Abbie
Abby
Abigael
Abigail
Abigale
Abra
Acacia
Ada
Adah
Adaline
Adar
>>> print(names.raw('male.txt')[:100])
Aamir
Aaron
Abbey
Abbie
Abbot
Abbott
Abby
Abdel
Abdul
Abdulkarim
Abdullah
Abe
Abel
Abelard
Abner
Abr

>>> names.words('female.txt')[:10]
['Abagael', 'Abagail', 'Abbe', 'Abbey', 'Abbi', 'Abbie', 'Abby', 'Abigael', 'Abigail', 'Abigale']
>>> names.words('female.txt')[-10:]
['Zonnya', 'Zora', 'Zorah', 'Zorana', 'Zorina', 'Zorine', 'Zsa Zsa', 'Zsazsa', 'Zulema', 'Zuzana']

#------------------------------------------------ Create two gendered name lists

>>> fnames = names.words('female.txt')
>>> mnames = names.words('male.txt')
```

```
>>> len(fnames)
5001
>>> len(mnames)
2943
>>> 'Zack' in mnames
True
>>> 'Zack' in fnames
False
>>> 'Taylor' in mnames
True
>>> 'Taylor' in fnames         # Wait, what? Gotta dig deeper
False
>>> print(names.readme())
Names Corpus, Version 1.3 (1994-03-29)
Copyright (C) 1991 Mark Kantrowitz
Additions by Bill Ross

This corpus contains 5001 female names and 2943 male names, sorted
alphabetically, one per line.

You may use the lists of names for any purpose, so long as credit is
given in any published work. You may also redistribute the list if you
provide the recipients with a copy of this README file. The lists are
not in the public domain (I retain the copyright on the lists) but are
freely redistributable.  If you have any additions to the lists of
names, I would appreciate receiving them.

Mark Kantrowitz <mkant+@cs.cmu.edu>
http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/nlp/corpora/names/

#---------------------------------------- Makes sense, data is from 1991...

>>> in_both = [n for n in mnames if n in fnames]
>>> len(in_both)
365
>>> in_both[:30]
['Abbey', 'Abbie', 'Abby', 'Addie', 'Adrian', 'Adrien', 'Ajay', 'Alex', 'Alexis', 'Alfie', 'Ali',
'Alix', 'Allie', 'Allyn', 'Andie', 'Andrea', 'Andy', 'Angel', 'Angie', 'Ariel', 'Ashley', 'Aubrey',
'Augustine', 'Austin', 'Averil', 'Barrie', 'Barry', 'Beau', 'Bennie', 'Benny']
>>> in_both[-30:]
['Timmy', 'Tobe', 'Tobie', 'Toby', 'Tommie', 'Tommy', 'Tony', 'Torey', 'Trace', 'Tracey', 'Tracie',
'Tracy', 'Val', 'Vale', 'Valentine', 'Van', 'Vin', 'Vinnie', 'Vinny', 'Virgie', 'Wallie', 'Wallis',
'Wally', 'Whitney', 'Willi', 'Willie', 'Willy', 'Winnie', 'Winny', 'Wynn']


#---------------------- What features will be useful? first char and last char

>>> def gender_features(word):
...     return {'firstchar': word[0], 'lastchar': word[-1]}
...
>>> gender_features('William')
{'firstchar': 'W', 'lastchar': 'm'}
>>> gender_features('Na-Rae')
{'firstchar': 'N', 'lastchar': 'e'}

#----------------------------------- Merging into a single list

>>> allnames = fnames + mnames
>>> allnames[:10]
['Abagael', 'Abagail', 'Abbe', 'Abbey', 'Abbi', 'Abbie', 'Abby', 'Abigael', 'Abigail', 'Abigale']
>>> allnames[-10:]
['Zed', 'Zedekiah', 'Zeke', 'Zelig', 'Zerk', 'Zeus', 'Zippy', 'Zollie', 'Zolly', 'Zorro']

#--------------------------------- Problem: we lost gender information with each name
#--------------------------------- Label each name with gender, THEN merge
```

```
>>> fnames_labeled = [(n, 'female') for n in fnames]
>>> mnames_labeled = [(n, 'male') for n in mnames]
>>> allnames_labeled = fnames_labeled + mnames_labeled
>>> allnames_labeled[:5]
[('Abagael', 'female'), ('Abagail', 'female'), ('Abbe', 'female'), ('Abbey', 'female'), ('Abbi',
'female')]
>>> allnames_labeled[-5:]
[('Zeus', 'male'), ('Zippy', 'male'), ('Zollie', 'male'), ('Zolly', 'male'), ('Zorro', 'male')]

#------------------------ Converting names into their feature representation

>>> allnames_feats = [(gender_features(n),g) for (n,g) in allnames_labeled]
>>> allnames_feats[:5]
[({'firstchar': 'A', 'lastchar': 'l'}, 'female'), ({'firstchar': 'A', 'lastchar': 'l'}, 'female'),
({'firstchar': 'A', 'lastchar': 'e'}, 'female'), ({'firstchar': 'A', 'lastchar': 'y'}, 'female'),
({'firstchar': 'A', 'lastchar': 'i'}, 'female')]
>>> allnames_feats[-5:]
[({'firstchar': 'Z', 'lastchar': 's'}, 'male'), ({'firstchar': 'Z', 'lastchar': 'y'}, 'male'),
({'firstchar': 'Z', 'lastchar': 'e'}, 'male'), ({'firstchar': 'Z', 'lastchar': 'y'}, 'male'),
({'firstchar': 'Z', 'lastchar': 'o'}, 'male')]

# ------------------------- Girl names front and boy names back --> must randomize!

>>> import random
>>> random.shuffle(allnames_feats)            # shuffles list IN PLACE


>>> allnames_feats[:5]
[({'firstchar': 'F', 'lastchar': 'd'}, 'female'), ({'firstchar': 'G', 'lastchar': 't'}, 'female'),
({'firstchar': 'B', 'lastchar': 'l'}, 'female'), ({'firstchar': 'C', 'lastchar': 'y'}, 'male'),
({'firstchar': 'F', 'lastchar': 'c'}, 'male')]
>>> allnames_feats[-5:]
[({'firstchar': 'C', 'lastchar': 'b'}, 'female'), ({'firstchar': 'S', 'lastchar': 'd'}, 'male'),
({'firstchar': 'S', 'lastchar': 'n'}, 'female'), ({'firstchar': 'S', 'lastchar': 'd'}, 'male'),
({'firstchar': 'D', 'lastchar': 's'}, 'male')]

#-------------------------- Partition feature list into test and train set

>>> test_set = allnames_feats[:500]
>>> train_set = allnames_feats[500:]
>>> len(test_set), len(train_set)
(500, 7444)

#------------------------------------------------ Now train a NB classifier

>>> boyorgirl = nltk.NaiveBayesClassifier.train(train_set)
>>> dir(boyorgirl)
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__',
'__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_feature_probdist', '_label_probdist',
'_labels', 'classify', 'classify_many', 'labels', 'most_informative_features', 'prob_classify',
'prob_classify_many', 'show_most_informative_features', 'train']
>>> boyorgirl.labels()
['male', 'female']

#----------------------------------------- Trying classifier on new names

>>> boyorgirl.classify('Neo')
Traceback (most recent call last):
  File "<pyshell#48>", line 1, in <module>
    boyorgirl.classify('Neo')
  File "C:\Program Files\Python311\Lib\site-packages\nltk\classify\naivebayes.py", line 89, in
classify
```

```
    return self.prob_classify(featureset).max()
  File "C:\Program Files\Python311\Lib\site-packages\nltk\classify\naivebayes.py", line 95, in
prob_classify
    featureset = featureset.copy()
AttributeError: 'str' object has no attribute 'copy'

             # Oops, can't directly classfy name string

>>> gender_features('Neo')
{'firstchar': 'N', 'lastchar': 'o'}
>>> boyorgirl.classify(gender_features('Neo'))             # classify on features
'male'
>>> boyorgirl.classify(gender_features('Na-Rae'))
'female'
>>> gender_features('Na-Rae')
{'firstchar': 'N', 'lastchar': 'e'}

#------------------------ Evaluating classifier's performance on test set

>>> nltk.classify.accuracy(boyorgirl, test_set)
0.816
>>> test_set[0]
({'firstchar': 'F', 'lastchar': 'd'}, 'female')
>>> test_set[1]
({'firstchar': 'G', 'lastchar': 't'}, 'female')

#------------------------------------ What are most informative features?

>>> boyorgirl.show_most_informative_features(30)
Most Informative Features
                lastchar = 'a'            female : male   =     37.2 : 1.0
                lastchar = 'k'              male : female =     32.9 : 1.0
                lastchar = 'f'              male : female =     15.8 : 1.0
                lastchar = 'p'              male : female =     11.8 : 1.0
                lastchar = 'm'              male : female =     11.1 : 1.0
                lastchar = 'd'              male : female =     10.1 : 1.0
                lastchar = 'v'              male : female =      9.1 : 1.0
                lastchar = 'o'              male : female =      7.7 : 1.0
                lastchar = 'r'              male : female =      6.6 : 1.0
                lastchar = 'w'              male : female =      5.8 : 1.0
                lastchar = 'g'              male : female =      5.2 : 1.0
               firstchar = 'W'              male : female =      5.1 : 1.0
                lastchar = 'z'              male : female =      4.6 : 1.0
                lastchar = 's'              male : female =      4.2 : 1.0
                lastchar = 't'              male : female =      4.0 : 1.0
                lastchar = 'j'              male : female =      3.9 : 1.0
                lastchar = 'i'            female : male   =      3.6 : 1.0
                lastchar = 'b'              male : female =      3.5 : 1.0
                lastchar = 'u'              male : female =      3.0 : 1.0
               firstchar = 'Q'              male : female =      2.6 : 1.0
               firstchar = 'U'              male : female =      2.5 : 1.0
               firstchar = 'K'            female : male   =      2.3 : 1.0
               firstchar = 'H'              male : female =      2.2 : 1.0
                lastchar = 'n'              male : female =      2.1 : 1.0
               firstchar = 'X'              male : female =      2.0 : 1.0
               firstchar = 'Z'              male : female =      1.9 : 1.0
                lastchar = 'x'              male : female =      1.9 : 1.0
                lastchar = 'e'            female : male   =      1.7 : 1.0
                lastchar = 'l'              male : female =      1.7 : 1.0
               firstchar = 'L'            female : male   =      1.7 : 1.0
```