```
import gensim
```

```
from gensim.models import Word2Vec # import word2vec library
```

```
# define training data for training
sentences = [['If','you','can','Dream','it','you','can','do','it'],
      ['Do','good','for','others','it','wil','come','back','in','unexpected','ways'],
      ['Being','happy','never','goes','out','of','style']]
```

```
#train word2vec model
model=Word2Vec(sentences,min_count=1)
```

```
print(model) #summrize the model
```

```
     Word2Vec(vocab=23, size=100, alpha=0.025)
```

```
#summrize vocubulary
word=list(model.wv.vocab)
```

```
print(word)
```

```
['If', 'you', 'can', 'Dream', 'it', 'do', 'Do', 'good', 'for', 'others', 'wil', 'come', 'back', 'in', 'unexpected', 'ways', 'Being', 'happy', 'never', 'goes
```

```
#Acess vector for one word
print(model['Dream'])
```

```
     [-1.5029708e-04 -2.9864330e-03  3.2388766e-03  3.3423428e-03
      -1.0855439e-03 -4.0680221e-03 -1.3953109e-03  4.4770916e-03
      -1.6865029e-03 -1.7473996e-03 -1.1032817e-03 -4.8777885e-03
      -1.5479167e-03  7.7229668e-04 -2.0218326e-03  3.3106606e-03
       3.4470488e-03  4.9211266e-03 -4.3372852e-03 -1.8739770e-03
```

```
 -2.3078513e-03 -1.0714391e-03 -1.5465165e-03 -3.7899821e-03
  3.9481353e-03  1.2945404e-04  4.5916163e-03 -5.4754334e-04
  3.1925645e-03 -3.1151280e-03  1.7865683e-03  1.8889360e-03
  4.9495241e-03 -4.6907645e-03 -4.2767371e-03 -3.1451718e-03
  5.1707699e-04  3.8931011e-03 -1.8055502e-03  3.2325049e-03
  1.1713954e-03 -8.8279363e-04 -9.1238006e-05  2.5157232e-03
 -8.6800486e-04  3.4882529e-03  1.1755255e-03 -2.0915021e-03
 -3.3863976e-03  2.6139527e-04 -3.4011208e-04 -4.3331748e-03
 -4.8604622e-03 -4.3736342e-03  3.3065358e-03  1.2865016e-03
 -5.4001203e-04  2.5513568e-03  3.8935139e-04  3.5672707e-03
  1.0568785e-03  5.6321023e-04 -2.5647972e-03 -2.7641968e-03
 -1.9874815e-03  1.0663944e-04  1.0909473e-06 -4.0298006e-03
  4.1907262e-03  1.0812311e-04 -2.0092356e-03 -1.2114734e-03
 -2.8699513e-03  3.0267084e-04  2.1096652e-03 -9.2331559e-04
 -1.8713290e-03  4.2444826e-03 -3.5290944e-03  1.9528902e-03
  1.4096743e-03  4.2235283e-03 -9.6797606e-04 -1.0577918e-03
  2.4282609e-03  2.6498011e-03  4.3948484e-03  2.6986431e-03
 -2.9869841e-03 -4.9542938e-03 -2.5189279e-03  2.9435512e-03
  2.9009490e-03 -3.8383058e-03 -7.5452507e-04  3.4474821e-03
 -1.0368370e-03  2.3017670e-03  4.0842062e-03 -1.7011535e-03]
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: DeprecationWarning: Call to deprecated `__getitem__` (Method will be removed in 4.0.0, use s
```

```
#save model
model.save('model.bin')
```

```
#load the model
new_model=Word2Vec.load("model.bin")
```

```
from gensim.models import KeyedVectors
filename = '/content/drive/MyDrive/GoogleNews-vectors-negative300.bin.gz'
model = KeyedVectors.load_word2vec_format(filename, binary=True)
```

```
result=model.most_similar(['good'])
# Now we could even use Word2vec to compute the similarity between two Make Models in the vocabulary by invoking the model.
# similarity( ) and passing in the relevant words.
```

```
print(result)
```

```
[('great', 0.7291510105133057), ('bad', 0.7190051078796387), ('terrific', 0.6889115571975708), ('decent', 0.6837348937988281), ('nice', 0.6836092472076416),
```

```
result1=model.most_similar("you")
print(result1)
```

```
[('You', 0.8077561855316162), ('your', 0.7808908820152283), ('yourself', 0.7698667645454407), ('I', 0.6739809513092041), ('we', 0.6565826535224915), ("Don't
```

```
result2=model.most_similar(['great'])
```

```
print(result2)
```

```
[('terrific', 0.7989331483840942), ('fantastic', 0.7935211658477783), ('tremendous', 0.7748856544494629), ('wonderful', 0.7647868394851685), ('good', 0.7291
```