```
import pandas as pd
import numpy as np
reviews_datasets = pd.read_csv("Reviews.csv")
reviews_datasets = reviews_datasets.head(20000)
reviews_datasets.dropna()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summa
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Go Qual Dog Fo
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not Advertis
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Deligl says it

```
reviews_datasets['Text'][350]
```

```
'These chocolate covered espresso beans are wonderful!  The chocolate is very dark and rich and the "bean" inside is a very delight
ful blend of flavors with just enough caffine to really give it a zing.'
```

```
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer(max_df=0.8, min_df=2, stop_words='english')
doc_term_matrix = count_vect.fit_transform(reviews_datasets['Text'].values.astype('U'))
```

```
print(doc_term_matrix)
```

```
  (0, 1076)     1
  (0, 8823)     1
  (0, 1338)     1
  (0, 2622)     1
  (0, 3352)     1
  (0, 6289)     1
  (0, 3630)     1
  (0, 6424)     1
  (0, 6287)     2
  (0, 4830)     1
  (0, 4729)     1
  (0, 7787)     1
  (0, 6276)     1
  (0, 5060)     1
  (0, 7456)     1
  (0, 924)      2
  (0, 3240)     1
  (0, 562)      1
  (1, 6287)     2
  (1, 603)      1
  (1, 4572)     1
  (1, 4456)     2
  (1, 7012)     1
  (1, 5860)     2
  (1, 295)      1
  :     :
  (8184, 7052)  2
  (8184, 316)   1
  (8184, 6518)  1
  (8184, 1833)  1
  (8184, 1573)  3
  (8184, 1769)  1
  (8184, 1273)  1
  (8184, 3363)  1
```

```
  (8184, 2953)  1
  (8184, 4074)  1
  (8184, 9078)  1
  (8184, 6021)  1
  (8184, 4896)  1
  (8185, 924)   1
  (8185, 3691)  1
  (8185, 5283)  1
  (8185, 5580)  1
  (8185, 9069)  1
  (8185, 9068)  2
  (8185, 2693)  2
  (8185, 5443)  1
  (8185, 6376)  1
  (8185, 2705)  1
  (8185, 3454)  1
  (8185, 2880)  1
```

```
import random
for i in range(10):
  random_id=random.randint(0,len(count_vect.get_feature_names_out()))
  print(count_vect.get_feature_names_out()[random_id])
```

```
gatorade
distortion
mistaken
sudden
outlet
dangerous
favourite
momma
energy
batters
```

```
from sklearn.decomposition import LatentDirichletAllocation
LDA = LatentDirichletAllocation(n_components=5, random_state=42)
LDA.fit(doc_term_matrix)
```

```
LatentDirichletAllocation
LatentDirichletAllocation(n_components=5, random_state=42)
```

```
first_topic=LDA.components_[0]
```

```
top_topic_words=first_topic.argsort()[-10:]
```

```
for i in top_topic_words:
  print(count_vect.get_feature_names_out()[i])
```

```
amazon
coffee
order
time
good
use
just
great
product
br
```

```
for i,topic in enumerate(LDA.components_):
  print(f"Top 10 words for topic #{i}:")
  print([count_vect.get_feature_names_out()[i] for i in topic.argsort()[-10:]])
  print('\n')
```

```
Top 10 words for topic #0:
['amazon', 'coffee', 'order', 'time', 'good', 'use', 'just', 'great', 'product', 'br']


Top 10 words for topic #1:
['love', 'taste', 'just', 'chocolate', 'product', 'like', 'mix', 'great', 'br', 'good']


Top 10 words for topic #2:
```

```
['best', 'organic', 'product', 'love', 'eat', 'amazon', 'like', 'dog', 'br', 'food']


Top 10 words for topic #3:
['just', 'orange', 'good', 'drink', 'juice', 'flavor', 'taste', 'like', 'chips', 'br']


Top 10 words for topic #4:
['sugar', 'just', 'cup', 'good', 'taste', 'flavor', 'like', 'tea', 'coffee', 'br']
```

```
topic_values=LDA.transform(doc_term_matrix)
topic_values.shape
```

```
(8187, 5)
```

```
reviews_datasets['Topic']=topic_values.argmax(axis=1)
```

```
reviews_datasets.head(10)
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine

Next steps: Generate code with reviews_datasets | View recommended plots | New interactive sheet

```
import pandas as pd
import numpy as np
reviews_datasets = pd.read_csv("Reviews.csv")
reviews_datasets = reviews_datasets.head(20000)
reviews_datasets.dropna()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	S
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Do
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Adv
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"[sa
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	M
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Gre

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vect = TfidfVectorizer(max_df=0.8, min_df=2, stop_words='english')
doc_term_matrix = tfidf_vect.fit_transform(reviews_datasets['Text'].values.astype('U'))
doc_term_matrix
```

```
<20000x14546 sparse matrix of type '<class 'numpy.float64'>'
	with 594703 stored elements in Compressed Sparse Row format>
```

```
from sklearn.decomposition import NMF
nmf = NMF(n_components=5, random_state=42)
nmf.fit(doc_term_matrix )
```

```
NMF
NMF(n_components=5, random_state=42)
```

```
import random
for i in range(10):
  random_id=random.randint(0,len(tfidf_vect.get_feature_names_out()))
  print(tfidf_vect.get_feature_names_out()[random_id])
```

```
vacationing
cakester
dehydrate
garlic
scientific
debating
chop
teacher
bears
gather
```

```
first_topic=nmf.components_[0]
top_topic_words=first_topic.argsort()[-10:]
```

```
for i in top_topic_words:
  print(tfidf_vect.get_feature_names_out()[i])
```

```
really
chocolate
love
```

```
flavor
just
product
taste
great
good
like
```

```
for i,topic in enumerate(nmf.components_):
  print(f"Top 10 words for topic #{i}:")
  print([tfidf_vect.get_feature_names_out()[i] for i in topic.argsort()[-10:]])
  print('\n')
```

```
Top 10 words for topic #0:
['really', 'chocolate', love', 'flavor', 'just', 'product', 'taste', 'great', 'good', 'like']


Top 10 words for topic #1:
['like', 'keurig', 'roast', 'flavor', 'blend', 'bold', 'strong', 'cups', 'cup', 'coffee']


Top 10 words for topic #2:
['com', 'amazon', 'orange', 'switch', 'water', 'drink', 'soda', 'sugar', 'juice', 'br']


Top 10 words for topic #3:
['bags', 'flavor', 'drink', 'iced', 'earl', 'loose', 'grey', 'teas', 'green', 'tea']


Top 10 words for topic #4:
['old', 'love', 'cat', 'eat', 'treat', 'loves', 'dogs', 'food', 'treats', 'dog']
```

```
from sklearn.decomposition import NMF
nmf = NMF(n_components=8, random_state=42)
nmf.fit(doc_term_matrix )
```

```
NMF
NMF(n_components=8, random_state=42)
```

```
for i in top_topic_words:
  print(tfidf_vect.get_feature_names_out()[i])
```

```
really
chocolate
love
flavor
just
product
taste
great
good
like
```

```
for i,topic in enumerate(nmf.components_):
  print(f"Top 10 words for topic #{i}:")
  print([tfidf_vect.get_feature_names_out()[i] for i in topic.argsort()[-15:]])
  print('\n')
```

```
Top 10 words for topic #0:
['peanut', 'butter', 'store', 'use', 'time', 'gluten', 'just', 'buy', 'free', 'love', 'amazon', 'price', 'good', 'product', 'great'


Top 10 words for topic #1:
['coffees', 'taste', 'good', 'keurig', 'smooth', 'bitter', 'like', 'roast', 'flavor', 'blend', 'bold', 'strong', 'cups', 'cup', 'co


Top 10 words for topic #2:
['know', 'ingredients', 'review', 'bag', 'pack', 'water', 'product', 'organic', 'href', 'gp', 'www', 'http', 'com', 'amazon', 'br']


Top 10 words for topic #3:
['taste', 'leaves', 'like', 'drink', 'black', 'love', 'bags', 'flavor', 'iced', 'earl', 'loose', 'grey', 'teas', 'green', 'tea']
```

```
Top 10 words for topic #4:
['ingredients', 'healthy', 'love', 'organic', 'newman', 'old', 'like', 'cat', 'eat', 'loves', 'treat', 'dogs', 'food', 'treats', 'do


Top 10 words for topic #5:
['natural', 'tangerine', 'kiwi', 'carbonated', 'fruit', 'flavor', 'switch', 'sweet', 'orange', 'taste', 'soda', 'like', 'sugar', 'dr


Top 10 words for topic #6:
['keurig', 'cups', 'mix', 'flavor', 'taste', 'tried', 'like', 'cookies', 'best', 'dark', 'cup', 'milk', 'cocoa', 'hot', 'chocolate'


Top 10 words for topic #7:
['spicy', 'salty', 'love', 'bags', 'flavors', 'snack', 'chip', 'vinegar', 'like', 'kettle', 'potato', 'bag', 'flavor', 'salt', 'chip
```

```
import pandas as pd
import numpy as np
reviews_datasets = pd.read_csv("Reviews.csv")
reviews_datasets = reviews_datasets.head(20000)
reviews_datasets.dropna()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	S
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Do
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Adv
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia	1	1	4	1219017600	"[

Start coding or generate with AI.