1. Step 1. read data
2. Step 2. packages
3. Step 3. DTM
4. Step 4. TFIDF matrix

In [1]:
```
import pandas as pd
```

In [2]:
```
# 1. read data
drRatings=pd.read_excel("OBGYN_new_train_80000.xlsx",nrows=100) # we use first 100 lines as an example
```

In [3]:
```
drRatings.head(5)
```

Out[3]:

	reviewID	doctorID	doctorName	specialty	numReviews	city	state	doctorHomepage	averageRating	staff	punctuality	helpf
0	1	2320644	Dr. Kevin G. Fahey	Gynecologist (OBGYN)	2	New Haven	MI	/doctor-ratings/2320644/Dr-KEVIN%2BG.-FAHEY-Ne...	4.25	4	3	
1	2	961169	Dr. Sudha R. Nair	Gynecologist (OBGYN)	5	Knoxville	TN	/doctor-ratings/961169/Dr-Sudha%2BR.-Nair-Knox...	2.00	2	3	
2	3	876934	Dr. Bonnie Gong	Gynecologist (OBGYN)	6	Kirkland	WA	/doctor-ratings/876934/Dr-Bonnie-Gong-Kirkland...	4.00	3	3	
3	4	102625	Dr. Louann Turner	Gynecologist (OBGYN)	6	Suffolk	VA	/doctor-ratings/102625/Dr-Louann-Turner-Suffol...	4.50	4	4	
4	5	42933	Dr. Michael A. Benson	Gynecologist (OBGYN)	21	Staten Island	NY	/doctor-ratings/42933/Dr-Michael%2BA.-Benson-S...	5.00	5	5	

In [4]:
```
# 3. DTM
from sklearn.feature_extraction.text import CountVectorizer
def calDTM(texts):
  vectorizer = CountVectorizer()
  DTM = vectorizer.fit_transform(texts)
  DTM=pd.DataFrame(DTM.toarray(),columns=vectorizer.get_feature_names())
  return(DTM)

#check the document of CountVectorizer for more parameters: https://scikit-learn.org/stable/modules/gene
```

In [5]:
```
## 3.1 DTM using raw text
DTM=calDTM(drRatings['review'])
DTM
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_fe
ature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please u
se get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[5]:

	10	10yrs	11p	11pm	12	120	13	15	150	15yrs	...	yeast	yes	yeung	you	younger	your	yrs	zero	ôv	œožud
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	2	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	4	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0	0	0	...	0	0	0	4	0	0	0	0	0	0
97	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
98	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
99	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

100 rows × 1428 columns

In [6]:
```
import re
from sklearn import feature_extraction
stop_words = feature_extraction.text.ENGLISH_STOP_WORDS
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer

def preprocess(text):
  text = text.lower() #lowercase
  text = re.sub(r'[^\w\s]', '', text) #remove punctuations
  text = re.sub(r'\d+', '', text) #remove numbers
  text = " ".join(text.split()) #stripWhitespace
  text = text.split()
  text = [x for x in text if x not in stop_words] #remove stopwords
  text = [x for x in text if x not in ["dr", "doctor"]] #remove task specific stopwords
  text = " ".join(text)
  # stemmer_ps = PorterStemmer()
  # text = [stemmer_ps.stem(word) for word in text.split()] #stemming
  # text = " ".join(text)
  # lemmatizer = WordNetLemmatizer()
  # text = [lemmatizer.lemmatize(word) for word in text.split()]  #lemmatization
  # text = " ".join(text)
  return(text)
```

In [7]:
```
## 3.2 DTM using preprocessed text
drRatings['text'] = drRatings['review'].apply(lambda x:preprocess(x))
print(drRatings['text'])
DTM=calDTM(drRatings['text'])
DTM
```

```
0     went close house great shares office doctors s...
1     unprofessional knowledgeable office surgical p...
2     leave practice illness disappointed try physic...
3     wonderful takes time tells exactly things term...
4     excellent caring considerate excellent bedside...
                            ...
95    completely trust believe b qualified capable c...
96    agree everthing person said lovell making feel...
97    kulwa nyc cares patients lot best gyn ive goog...
98    unbelievable extremely high risk pregnancys wo...
99    just myomectomy salvay hoped better outcome ab...
Name: text, Length: 100, dtype: object

/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_fe
ature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please u
se get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[7]:

	_x__x__x_her	aa	abbrasive	abhorrence	abilities	ablation	able	abnormal	absolute	absolutely	...	years	yeast	yes	yeung	youll
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
96	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
97	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
98	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
99	0	0	0	1	0	0	1	0	0	0	...	1	0	0	0	0

100 rows × 1219 columns

In [8]:
```
# 4. tf.idf matrix
from sklearn.feature_extraction.text import TfidfVectorizer

def calTFIDF(texts,max_features=None):
  vectorizer = TfidfVectorizer(max_features=max_features)
  TFIDF = vectorizer.fit_transform(texts)
  TFIDF=pd.DataFrame(TFIDF.toarray(),columns=vectorizer.get_feature_names())
  return(TFIDF)

#check the document of TfidfVectorizer for more parameters: https://scikit-learn.org/stable/modules/gene
```

In [9]:
```
TFIDF=calTFIDF(drRatings['text'])
TFIDF
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_fe
ature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please u
se get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[9]:

	_x__x__x_her	aa	abbrasive	abhorrence	abilities	ablation	able	abnormal	absolute	absolutely	...	years	yeast	yes	yeur
0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
1	0.0	0.0	0.0	0.000000	0.0	0.370655	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
2	0.0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
3	0.0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
4	0.0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	0.0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
96	0.0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
97	0.0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
98	0.0	0.0	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.0	...	0.000000	0.0	0.0	0
99	0.0	0.0	0.0	0.129597	0.0	0.000000	0.111347	0.0	0.0	0.0	...	0.073249	0.0	0.0	0

100 rows × 1219 columns

In [10]:
```
# 5. keep only top terms as our vocabulary
print(TFIDF.shape)  # check the number of unique terms -- which is the maximum size of our vocabulary
TFIDF=calTFIDF(drRatings['text'],max_features=6)
TFIDF
```

```
(100, 1219)

/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_fe
ature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please u
se get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

Out[10]:

	baby	care	great	office	staff	time
0	0.0	0.0	0.969051	0.246861	0.0	0.0
1	0.0	0.0	0.000000	1.000000	0.0	0.0
2	0.0	0.0	0.000000	0.000000	0.0	1.0
3	0.0	0.0	0.000000	0.000000	0.0	1.0
4	0.0	0.0	0.000000	0.000000	0.0	0.0
...	...	...	...	...	...	...
95	0.0	0.0	0.000000	0.000000	0.0	0.0
96	1.0	0.0	0.000000	0.000000	0.0	0.0
97	0.0	0.0	0.000000	0.000000	0.0	0.0
98	0.0	0.0	0.000000	0.000000	0.0	0.0
99	0.0	0.0	0.000000	0.000000	0.0	0.0

100 rows × 6 columns

In [ ]:

In [ ]: