Reviews Part IV: NLTK Top Words in All Comments

In [21]:

```
import pandas as pd
from textblob import TextBlob
from wordcloud import WordCloud
```

In [22]:

```
cols = ["comments"]
reviews= pd.read_csv("reviews.csv", usecols = cols)
reviews["comments"] = reviews["comments"].astype(str) #put the comments into strings
reviews.head()
```

Out[22]:

	comments
0	My girlfriend and I hadn't known Alina before ...
1	Alina was a really good host. The flat is clea...
2	Alina is an amazing host. She made me feel rig...
3	Alina's place is so nice, the room is big and ...
4	Nice location in Islington area, good for shor...

In [23]:

```
reviews.comments.head()
```

Out[23]:

```
0    My girlfriend and I hadn't known Alina before ...
1    Alina was a really good host. The flat is clea...
2    Alina is an amazing host. She made me feel rig...
3    Alina's place is so nice, the room is big and ...
4    Nice location in Islington area, good for shor...
Name: comments, dtype: object
```

In [24]:

```
reviews.comments.values[1]
```

Out[24]:

```
'Alina was a really good host. The flat is clean and tidy - and really close to Finsbury Park station which is
quite close to Central London. I recommend Alina to everyone. '
```

Import Natural Language Processing Libraries

In [25]:

```
#Natural Language processing
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import re
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from wordcloud import WordCloud
```

Preprocessing of the reviews data

In [27]:

```
#Preprocessing of the data
reviews = reviews[reviews['comments'].notnull()]
                                       #Take out empty comments
reviews['comments'] = reviews['comments'].str.replace('\d+','')
                                       #remove numbers
reviews['comments'] = reviews['comments'].str.lower()
                                       #lowercase
reviews['comments'] = reviews['comments'].str.replace('\r\n',"")
                                       #remove windows new line
stop_english=stopwords.words("english")
reviews['comments'] = reviews['comments'].apply(lambda x: " ".join([i for i in x.split()
                                                    if i not in (stop_english)]))
                                       #remove all the stop words with nltk library
reviews['comments'] = reviews['comments'].str.replace('[^\w\s]'," ")
                                       #remove all punctuation
reviews['comments'] = reviews['comments'].str.replace('\s+', ' ')
                                       #replace x spaces by one space



reviews['comments'].values[1]
                                       #print the comment index1 one more time
```

```
/var/folders/0k/qsrs17bs5n1gr22p3vddn0xm0000gn/T/ipykernel_1382/1975952342.py:4: FutureWarning: The default val
ue of regex will change from True to False in a future version.
  reviews['comments'] = reviews['comments'].str.replace('\d+','')
/var/folders/0k/qsrs17bs5n1gr22p3vddn0xm0000gn/T/ipykernel_1382/1975952342.py:14: FutureWarning: The default va
lue of regex will change from True to False in a future version.
  reviews['comments'] = reviews['comments'].str.replace('[^\w\s]'," ")
/var/folders/0k/qsrs17bs5n1gr22p3vddn0xm0000gn/T/ipykernel_1382/1975952342.py:16: FutureWarning: The default va
lue of regex will change from True to False in a future version.
  reviews['comments'] = reviews['comments'].str.replace('\s+', ' ')
```

Out[27]:

```
'alina really good host flat clean tidy really close finsbury park station quite close central london recommend
alina everyone '
```

Top 10 common words in the comments

In [30]:

```
#Top 10 common words in the comments with CountVectorizer()
texts= reviews.comments.tolist()

vec = CountVectorizer().fit(texts)
bag_of_words = vec.transform(texts)
sum_words = bag_of_words.sum(axis=0)
words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]

cvec_df = pd.DataFrame.from_records(words_freq, columns= ['words', 'counts']).sort_values(by="counts", ascendir
cvec_df.head(10)
```

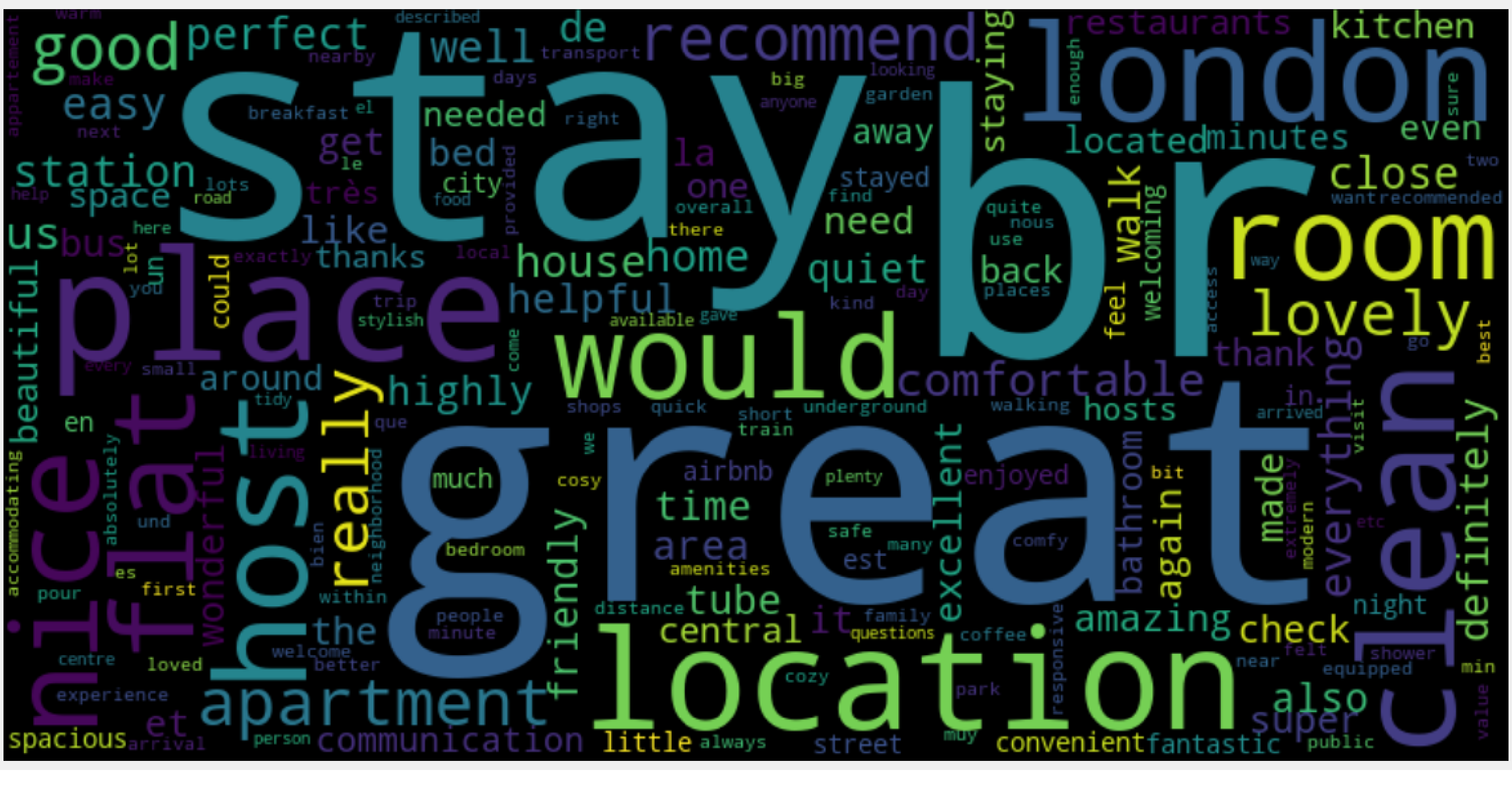
Out[30]:

	words	counts
149	br	636149
113	great	494543
56	stay	438999
25	place	385157
132	location	320683
51	london	290717
75	clean	280026
74	host	259986
108	nice	230012
109	room	219878

In [29]:

```
#Create the word cloud from the file we have
cvec_dict = dict(zip(cvec_df.words, cvec_df.counts))

wordcloud = WordCloud(width=800, height=400)
wordcloud.generate_from_frequencies(frequencies=cvec_dict)
plt.figure( figsize=(20,10) )
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```


stay
br
great
place
location
london
clean
host
nice
room
good
perfect
well
de
recommend
restaurants
kitchen
easy
needed
bed
away
located
minutes
even
station
space
city
stayed
close
us
bus
like
thanks
house
home
need
quiet
back
lovely
helpful
would
comfortable
thank
really
highly
around
beautiful
flat
wonderful
much
time
airbnb
area
tube
central
friendly
excellent
enjoyed
bathroom
again
made
everything
definitely
amazing
check
night
apartment
little
street
convenient
fantastic
also
super
spacious
communication
walk
feel
welcoming
staying
could
everyone
get
one
la
et
en
the