```
import pandas as pd
import re
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from nltk.stem import PorterStemmer
import nltk

nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

```
df=pd.read_csv("/content/data.csv")
```

```
df
```

	review	rating
0	It was nice produt. I like it's design a lot. ...	5
1	awesome sound....very pretty to see this nd th...	5
2	awesome sound quality. pros 7-8 hrs of battery...	4
3	I think it is such a good product not only as ...	5
4	awesome bass sound quality very good bettary l...	5
...	...	...
9971	GoodREAD MORE	5
9972	Everything is amazimg but the built is very li...	5
9973	GoodREAD MORE	5
9974	Best headphone i have ever used....READ MORE	5
9975	NiceREAD MORE	5

9976 rows × 2 columns

```
df=df[['review']].dropna()
```

```
df
```

	review
0	It was nice produt. I like it's design a lot. ...
1	awesome sound....very pretty to see this nd th...
2	awesome sound quality. pros 7-8 hrs of battery...
3	I think it is such a good product not only as ...
4	awesome bass sound quality very good bettary l...
...	...
9971	GoodREAD MORE
9972	Everything is amazimg but the built is very li...
9973	GoodREAD MORE
9974	Best headphone i have ever used....READ MORE
9975	NiceREAD MORE

9976 rows × 1 columns

```
df.isnull().sum()
```

```
review    0
dtype: int64
```

```
df.reset_index(drop=True, inplace=True)
```

```
df
```

	review
0	It was nice produt. I like it's design a lot. ...
1	awesome sound....very pretty to see this nd th...
2	awesome sound quality. pros 7-8 hrs of battery...
3	I think it is such a good product not only as ...
4	awesome bass sound quality very good bettary l...
...	...
9971	GoodREAD MORE
9972	Everything is amazimg but the built is very li...
9973	GoodREAD MORE
9974	Best headphone i have ever used....READ MORE
9975	NiceREAD MORE

9976 rows × 1 columns

```
X=df['review']
```

```
import nltk
nltk.download('stopwords')
ps =PorterStemmer()
corpus=[]
for i in range(len(X)):
    review=re.sub("[^a-zA-Z]"," ",X[i])
    review=review.lower()
    review=review.split()
    review=[ps.stem(word) for word in review if word not in set(stopwords.words("english"))]
    review=" ".join(review)
    corpus.append(review)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
import nltk
nltk.download('stopwords')

ps =PorterStemmer()

def preprocess_text(text):
    review=re.sub("[^a-zA-Z]"," ",text)
    print(review)
    review=review.lower()
    review=review.split()
    review=[ps.stem(word) for word in review if word not in set(stopwords.words("english"))]
    review=" ".join(review)
    return review
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
Start coding or generate with AI.
```

```
df=df[['review']].dropna()
```

```
df['cleaned_data'] = df['review'].apply(preprocess_text)
```

```
df
```

	review	cleaned_data
0	It was nice produt. I like it's design a lot. ...	nice produt like design lot easi carri look st...
1	awesome sound....very pretty to see this nd th...	awesom sound pretti see nd sound qualiti good ...
2	awesome sound quality. pros 7-8 hrs of battery...	awesom sound qualiti pro hr batteri life inclu...
3	I think it is such a good product not only as ...	think good product per qualiti also design qui...
4	awesome bass sound quality very good bettary l...	awesom bass sound qualiti good bettari long li...
...	...	...

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation

# Vectorization
tfidf_vectorizer = TfidfVectorizer(max_features=1000)
tfidf_matrix = tfidf_vectorizer.fit_transform(df['cleaned_data'])

# Topic Modeling (LDA)
num_topics = 5
lda_model = LatentDirichletAllocation(n_components=num_topics, random_state=42)
lda_model.fit(tfidf_matrix)

# Print the topics and their top keywords
def print_top_words(model, feature_names, n_top_words):
    for topic_idx, topic in enumerate(model.components_):
        print(f"Topic #{topic_idx}:")
        print(" ".join([feature_names[i] for i in topic.argsort()[:-n_top_words - 1:-1]]))
        print()

print_top_words(lda_model, tfidf_vectorizer.get_feature_names_out(), 10)
```

```
Topic #0:
good work use bluetooth read sound awesomeread connect qualiti headphon

Topic #1:
productread nice good niceread read product awesom super moneyread valu

Topic #2:
best product read good amaz sound bass qualiti thank headphon

Topic #3:
goodread ear qualityread good sound pain qualiti tight comfort use

Topic #4:
product read love good excel qualiti itread sound superread price
```

```
Start coding or generate with AI.
```