```
from scipy.stats import kruskal
import matplotlib.pyplot as plt
import nltk
nltk.download('stopwords')
```

```
import pandas as pd
import numpy as np
import random
import seaborn as sns
import matplotlib.pyplot as plt
```

```
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
import re
import string
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Topic Modeling

```
!pip install gensim
```

```
Requirement already satisfied: gensim in /usr/local/lib/python3.10/dist-packages (4.3.2)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.23.5)
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.11.4)
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from gensim) (6.4.0)
```

```
from gensim import corpora, models
from gensim.models import LdaModel
from gensim.parsing.preprocessing import preprocess_string
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
```

Topic Modeling Based on Star Ratings

```
%%capture
!pip install bertopic
```

```
import pandas as pd
df = pd.read_csv('topicModelling.csv', encoding="ISO-8859-1")
df.head()
```

	review	rating	cleanReview	reviewForBERT
0	Why does it look like someone spit on my food?...	1	whi doe look like someon spit food normal tran...	Why does it look like someone spit on my food\...
1	It'd McDonalds. It is what it is as far as the...	4	itd mcdonald far food atmospher go staff doe m...	Itd McDonalds It is what it is as far as the f...
2	Made a mobile order got to the speaker and che...	1	made mobil order got speaker check line wa mov...	Made a mobile order got to the speaker and che...
3	My mc. Crispy chicken sandwich was	[illegible]	mc crispi chicken sandwich wa custom servic	My mc Crispy chicken sandwich was customer

```
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS

def remove_stop_words(text):
    if isinstance(text, str):
        stop_words = set(stopwords.words('english')) | ENGLISH_STOP_WORDS
        words = text.split()
        filtered_words = [word.lower() for word in words if word.lower() not in stop_words]
        return ' '.join(filtered_words)
    else:
        return ''  # or return some default value if the input is not a string

# Apply the function to the 'reviewForBERT' column and create a new column 'reviewWithoutStopWords'
df['reviewWithoutStopWords'] = df['reviewForBERT'].apply(remove_stop_words)

# Display the updated DataFrame
print(df)
```

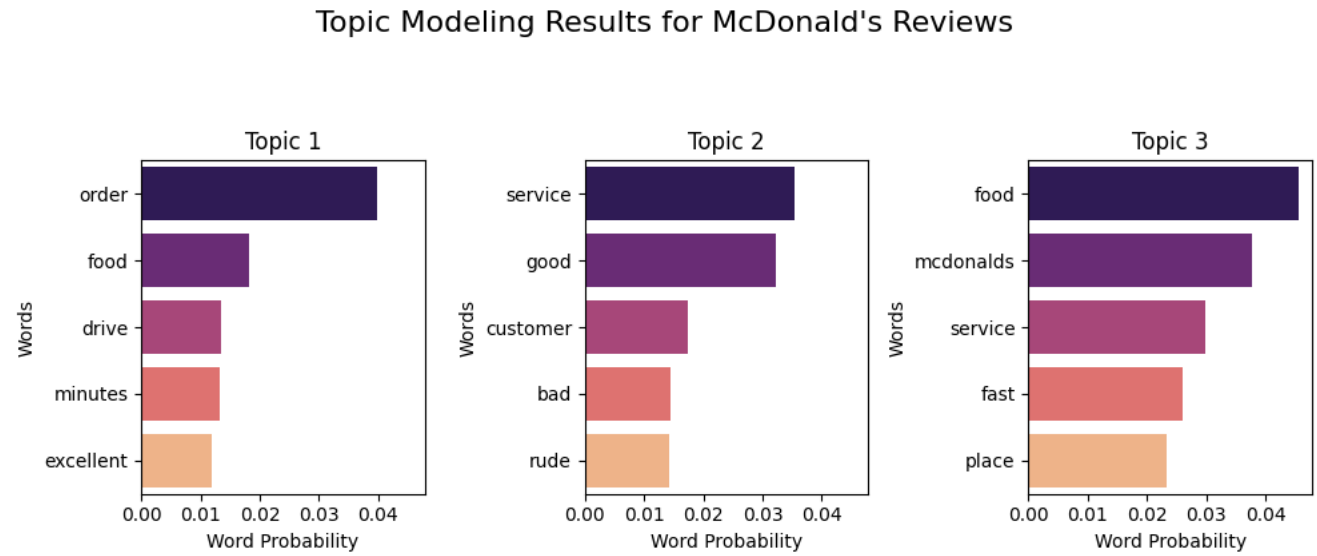
```
                                                   review  rating  \
0      Why does it look like someone spit on my food?...       1
1      It'd McDonalds. It is what it is as far as the...       4
2      Made a mobile order got to the speaker and che...       1
3      My mc. Crispy chicken sandwich was Ã¯Â¿Â½Ã¯Â¿Â...       5
4      I repeat my order 3 times in the drive thru, a...       1
...                                                  ...     ...
33391                        They treated me very badly.       1
33392                           The service is very good       5
33393                         To remove hunger is enough       4
33394  It's good, but lately it has become very expen...       5
33395                          they took good care of me       5

                                            cleanReview  \
0      whi doe look like someon spit food normal tran...
1      itd mcdonald far food atmospher go staff doe m...
2      made mobil order got speaker check line wa mov...
3      mc crispi chicken sandwich wa custom servic wa...
4      repeat order 3 time drive thru still manag mes...
...                                                  ...
33391                                   treat veri badli
33392                                   servic veri good
33393                                 remov hunger enough
33394                     good late ha becom veri expens
33395                                     took good care

                                          reviewForBERT  \
0      Why does it look like someone spit on my food\...
1      Itd McDonalds It is what it is as far as the f...
2      Made a mobile order got to the speaker and che...
3      My mc Crispy chicken sandwich was  customer se...
4      I repeat my order 3 times in the drive thru an...
...                                                  ...
33391                         They treated me very badly
33392                           The service is very good
33393                         To remove hunger is enough
33394   Its good but lately it has become very expensive
33395                          they took good care of me

                                reviewWithoutStopWords
0      look like spit food normal transaction chill p...
1      itd mcdonalds far food atmosphere staff make d...
2      mobile order got speaker checked line moving l...
3      mc crispy chicken sandwich customer service qu...
4      repeat order 3 times drive manage mess suppose...
...                                                  ...
33391                                     treated badly
33392                                      service good
33393                                     remove hunger
33394                              good lately expensive
33395                                     took good care

[33396 rows x 5 columns]
```

```
#Identifying top 3 topics in each star rating with their review samples
# Function to perform topic modeling
def perform_topic_modeling(df, num_topics=3, num_sample_reviews=3):
    # Tokenize the text
    tokenized_text = df['reviewWithoutStopWords'].apply(lambda x: x.split())

    # Create a dictionary representation of the documents
    dictionary = corpora.Dictionary(tokenized_text)

    # Convert tokenized documents into a document-term matrix
    corpus = [dictionary.doc2bow(text) for text in tokenized_text]

    # Train LDA model
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=15, random_state=42)

    # Display the topics
    for topic_idx in range(num_topics):
        print(f"\nTopic #{topic_idx + 1}: {lda_model.print_topic(topic_idx)}")

        # Print three random reviews for each topic
        print("Sample Reviews:")
        sample_reviews = df.sample(num_sample_reviews)['reviewWithoutStopWords']
        for i, review in enumerate(sample_reviews):
            print(f"{i + 1}. {review}")
        print()

# Perform topic modeling for each rating category
print("1 - Star Rating")
perform_topic_modeling(df[df['rating'] == 1])
print("2 - Star Rating")
perform_topic_modeling(df[df['rating'] == 2])
print("3 - Star Rating")
perform_topic_modeling(df[df['rating'] == 3])
print("4 - Star Rating")
perform_topic_modeling(df[df['rating'] == 4])
print("5 - Star Rating")
perform_topic_modeling(df[df['rating'] == 5])
```

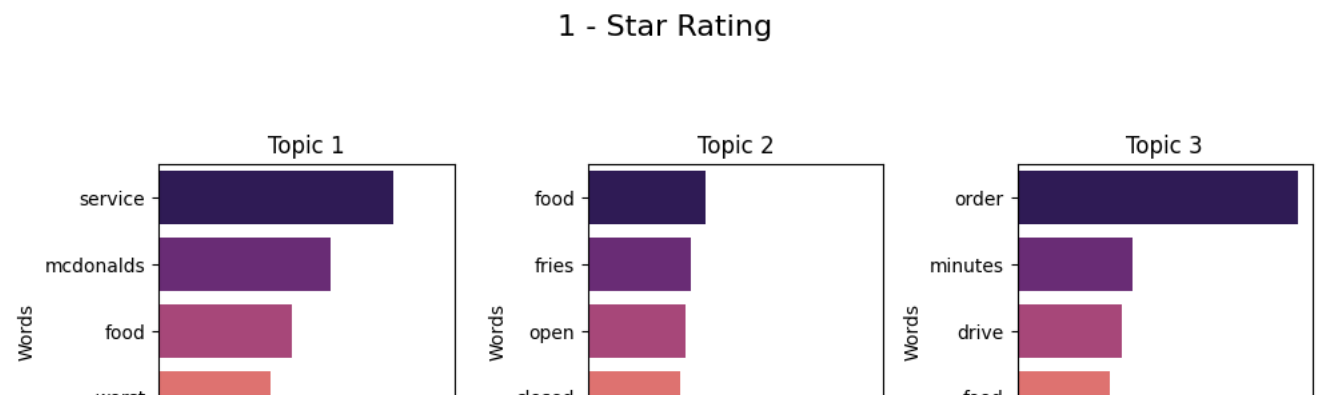
```
1 - Star Rating

Topic #1: 0.035*"service" + 0.025*"mcdonalds" + 0.020*"food" + 0.017*"worst" + 0.015*"place" + 0.014*"dont" + 0.013*"staff"
Sample Reviews:
1. ordered bacon egg cheese biscuit meal bacon send 2 complimentary meals great free coupon gave chicken sandwich ordered si
2. ordered times mcdonalds screw cheeseburger asked ketchup mustard pickles onion got condiments meat mean really learn read
3. restroom bad service


Topic #2: 0.017*"food" + 0.015*"fries" + 0.014*"open" + 0.014*"closed" + 0.011*"hours" + 0.011*"like" + 0.009*"place" + 0.00
Sample Reviews:
1. bad service took 30 minutes food cashier forget son food wait food 15 minutes
2. want apologize recommended particular mcdonalds past restaurant used recommend tourists looking fast place eat away disne
3. terrible


Topic #3: 0.042*"order" + 0.017*"minutes" + 0.015*"drive" + 0.014*"food" + 0.011*"got" + 0.011*"asked" + 0.010*"ordered" + 0
Sample Reviews:
1. sat drive tuesday night 1130 pm 20 minutes pull order associate says sorry closed explained waiting 20 minutes mumbled sa
2. tow car 430
3. unprofessional worker long make food

2 - Star Rating

Topic #1: 0.019*"order" + 0.017*"food" + 0.016*"mcdonalds" + 0.013*"minutes" + 0.011*"drive" + 0.008*"time" + 0.006*"didnt"
Sample Reviews:
1. poor
2. pretty employee unhelpful clearly unhappy order wrong time
3. didnt eat time close lobby 10pm didnt drivethru cause im motorcycle kind dingus closes half store customers old town


Topic #2: 0.024*"food" + 0.023*"service" + 0.021*"order" + 0.015*"mcdonalds" + 0.015*"slow" + 0.012*"good" + 0.011*"time" +
Sample Reviews:
1. mcdonalds ive security guard let pee order place clean fine
2. using kiosk selections choose delete things order got frustrated busy counter personnel answer question informed customer
3. slow slow slow younger people rude new renovation slow workers rudeness past saturday waited 15 min drive placed order 10


Topic #3: 0.022*"poor" + 0.019*"fries" + 0.018*"food" + 0.017*"order" + 0.013*"cold" + 0.013*"got" + 0.012*"like" + 0.012*"d
Sample Reviews:
1. poor
2. went drivethru preorder rushed busy took bite sandwiches completely wrong worth going happened multiple times location
3. cup fingered cashier stars gave new asked
```

```
3 - Star Rating

Topic #1: 0.042*"food" + 0.033*"mcdonalds" + 0.033*"service" + 0.024*"good" + 0.018*"fast" + 0.017*"slow" + 0.017*"drive" +
Sample Reviews:
1. decent mcdonalds recently remodeled clean new tend attract transient population
2. neutral
3. service nice feel comfortable people sitting table eating like family


Topic #2: 0.065*"neutral" + 0.019*"order" + 0.013*"food" + 0.011*"dont" + 0.009*"mcdonalds" + 0.008*"like" + 0.006*"ice" + 0
Sample Reviews:
1. slow service drive 21 min waiting time car ahead 11pm
2. neutral
3. ok
```

```
#visualizing those top 3 topics in each star ratings
def perform_topic_modeling(df,graph_title, num_topics=3, num_sample_reviews=3):

    # Tokenize the text
    tokenized_text = df['reviewWithoutStopWords'].apply(lambda x: x.split())

    # Create a dictionary representation of the documents
    dictionary = corpora.Dictionary(tokenized_text)

    # Convert tokenized documents into a document-term matrix
    corpus = [dictionary.doc2bow(text) for text in tokenized_text]

    # Train LDA model
    lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=15, random_state=42)

    topics = lda_model.show_topics(num_topics=lda_model.num_topics, num_words=5, formatted=False)

    fig, axes = plt.subplots(1, lda_model.num_topics, figsize=(10, 4), sharex=True)
    axes = axes.flatten()

    for i, (topic, ax) in enumerate(zip(topics, axes)):
        topic_words = [word for word, _ in topic[1]]
        word_probs = [count for _, count in topic[1]]

        sns.barplot(x=word_probs, y=topic_words, ax=ax, palette='magma')
        ax.set_title(f'Topic {i + 1}')
        ax.set_xlabel('Word Probability')
        ax.set_ylabel('Words')

    plt.suptitle(graph_title, y=1.05, fontsize=16)
    plt.tight_layout(rect=[0, 0, 1, 0.95])  # Adjust the layout to prevent the title from being cut off
    plt.show()

# Example usage:
perform_topic_modeling(df, num_topics=3, num_sample_reviews=3, graph_title="Topic Modeling Results for McDonald's Reviews")


# Perform topic modeling for each rating category
print("1 - Star Rating")
perform_topic_modeling(df[df['rating'] == 1],graph_title = '1 - Star Rating')
print("2 - Star Rating")
perform_topic_modeling(df[df['rating'] == 2], graph_title = '2 - Star Rating')
print("3 - Star Rating")
perform_topic_modeling(df[df['rating'] == 3], graph_title = '3 - Star Rating')
print("4 - Star Rating")
perform_topic_modeling(df[df['rating'] == 4], graph_title = '4 - Star Rating')
print("5 - Star Rating")
perform_topic_modeling(df[df['rating'] == 5], graph_title = '5 - Star Rating')
```




```
1 - Star Rating
```



Bert Topic Modeling:

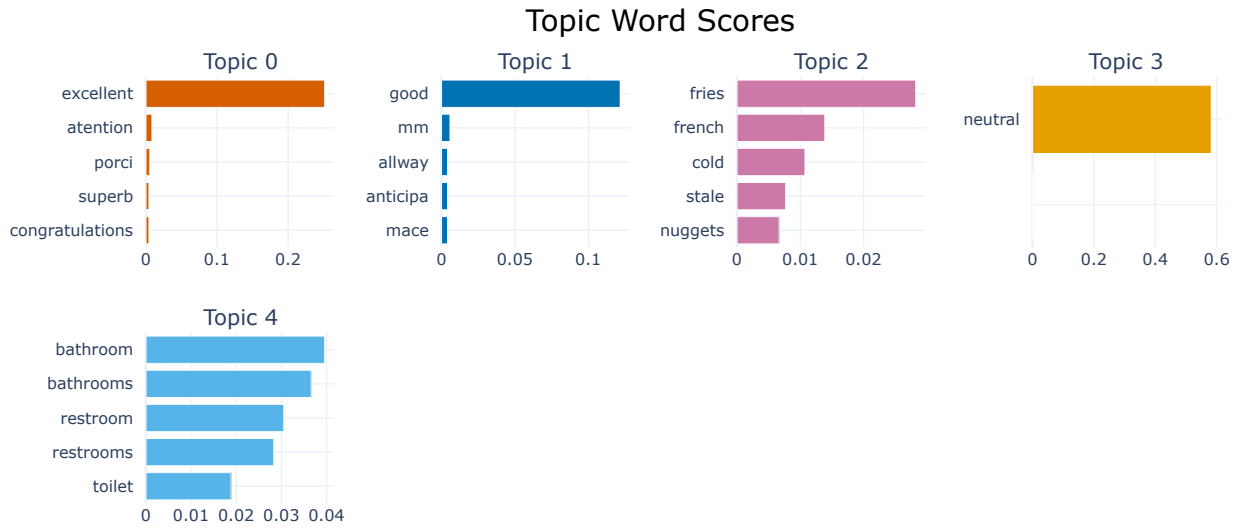
```
#Performed Bert topic modeling for further analysis
from bertopic import BERTopic
from sentence_transformers import SentenceTransformer
```

```
# Use a Sentence Transformer model for embedding sentences
model = SentenceTransformer('paraphrase-MiniLM-L6-v2')

# Create BERTopic model
bertopic_model = BERTopic(language="english", embedding_model=model)


# Handle missing values
df['reviewWithoutStopWords'] = df['reviewWithoutStopWords'].fillna('')

# Fit the model on the review data
topics, _ = bertopic_model.fit_transform(df['reviewWithoutStopWords'])

# Visualize the topics
bertopic_model.visualize_topics()
```

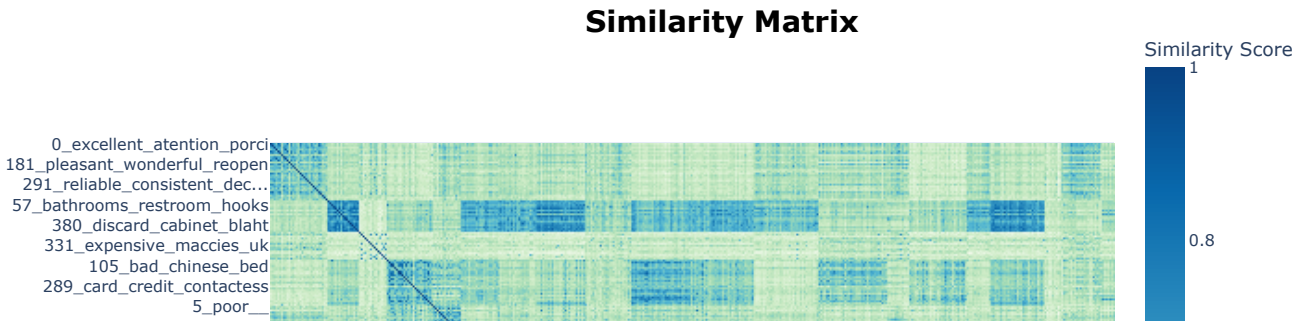
```
.gitattributes: 100%                                690/690 [00:00<00:00, 48.9kB/s]
1_Pooling/config.json: 100%                         190/190 [00:00<00:00, 15.2kB/s]
README.md: 100%                                     3.69k/3.69k [00:00<00:00, 219kB/s]
config.json: 100%                                   629/629 [00:00<00:00, 41.2kB/s]
config_sentence_transformers.json: 100%             122/122 [00:00<00:00, 9.50kB/s]
pytorch_model.bin: 100%                             90.9M/90.9M [00:00<00:00, 174MB/s]
sentence_bert_config.json: 100%                     53.0/53.0 [00:00<00:00, 4.11kB/s]
special_tokens_map.json: 100%                       112/112 [00:00<00:00, 6.12kB/s]
tokenizer.json: 100%                                466k/466k [00:00<00:00, 3.39MB/s]
tokenizer_config.json: 100%                         314/314 [00:00<00:00, 13.3kB/s]
vocab.txt: 100%                                     232k/232k [00:00<00:00, 1.74MB/s]
modules.json: 100%                                  229/229 [00:00<00:00, 14.8kB/s]
```

```
bertopic_model.visualize_hierarchy(top_n_topics=30)
```

```
bertopic_model.visualize_barchart(top_n_topics=5)
```




```
bertopic_model.visualize_heatmap(n_clusters=20, width=1000, height=1000)
```


Similarity Matrix
Similarity Score
1
0.8
0_excellent_atention_porci
181_pleasant_wonderful_reopen
291_reliable_consistent_dec...
57_bathrooms_restroom_hooks
380_discard_cabinet_blaht
331_expensive_maccies_uk
105_bad_chinese_bed
289_card_credit_contactess
5_poor__