



## EXPERIMENT NO -05

### CODE :

```
# Install necessary libraries
!pip install nltk spacy gensim wordcloud pyLDAvis bokeh

# Import libraries
import nltk
import re
import numpy as np
import pandas as pd
import gensim
import spacy
import logging
import warnings
import gensim.corpora as corpora
import matplotlib.pyplot as plt

from pprint import pprint
from nltk.corpus import stopwords
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel
from wordcloud import WordCloud, STOPWORDS
import matplotlib.colors as mcolors
from sklearn.manifold import TSNE
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import pyLDAvis
import pyLDAvis.gensim

# Download stopwords
nltk.download('stopwords')
stop_words = stopwords.words('english')

# Load dataset
df = pd.read_csv('/content/google_reviews.csv')

# Preprocessing function
def sent_to_words(sentences):
    for sent in sentences:
        sent = re.sub('\s+', ' ', sent) # Remove newline characters
        sent = re.sub('"', '', sent) # Remove single quotes
        sent = gensim.utils.simple_preprocess(str(sent), deacc=True)
        yield sent

# Convert to list
All_reviews = df['review_text'].values.tolist()
reviews_words = list(sent_to_words(All_reviews))
```



```
# Build bigram and trigram models
# Build bigram and trigram models
bigram = gensim.models.Phrases(reviews_words, min_count=5, threshold=10)
trigram = gensim.models.Phrases(bigram[reviews_words], threshold=10)
# Use gensim.models.phrases.Phraser directly
bigram_mod = gensim.models.phrases.Phraser(bigram)
trigram_mod = gensim.models.phrases.Phraser(trigram)

# Function for stopwords removal, bigrams, trigrams, and lemmatization
def process_words(texts, stop_words=stop_words, allowed_postags=['NOUN', 'ADJ',
'VERB', 'ADV']):
    texts = [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc
in texts]
    texts = [bigram_mod[doc] for doc in texts]
    texts = [trigram_mod[bigram_mod[doc]] for doc in texts]

    nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner'])
    texts_out = []

    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])

    # Remove stopwords again after lemmatization
    texts_out = [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for
doc in texts_out]
    return texts_out

# Processed reviews
data_final = process_words(reviews_words)

# Create Dictionary and Corpus
id2word = corpora.Dictionary(data_final)
corpus = [id2word.doc2bow(text) for text in data_final]

# Build LDA Model
lda_model = gensim.models.LdaModel(
    corpus=corpus,
    id2word=id2word,
    num_topics=7,
    random_state=100,
    update_every=1,
    chunksize=10,
    passes=10,
    alpha='symmetric',
    iterations=100,
    per_word_topics=True
)

# Print the topics
```



```
pprint(lda_model.print_topics())

# Generate Word Cloud
cols = [color for name, color in mcolors.TABLEAU_COLORS.items()]
cloud = WordCloud(
    stopwords=stop_words,
    background_color='white',
    width=2500,
    height=1800,
    max_words=10,
    colormap='tab10'
)

topics = lda_model.show_topics(formatted=False)
fig, axes = plt.subplots(3, 2, figsize=(10, 10), sharex=True, sharey=True)

for i, ax in enumerate(axes.flatten()):
    fig.add_subplot(ax)
    topic_words = dict(topics[i][1])
    cloud.generate_from_frequencies(topic_words, max_font_size=300)
    plt.gca().imshow(cloud)
    plt.gca().set_title('Topic ' + str(i), fontdict=dict(size=16))
    plt.gca().axis('off')

plt.subplots_adjust(wspace=0, hspace=0)
plt.axis('off')
plt.margins(x=0, y=0)
plt.tight_layout()
plt.show()

# Get topic weights
topic_weights = []
for i, row_list in enumerate(lda_model[corpus]):
    topic_weights.append([w for i, w in row_list[0]])

# Convert to numpy array
arr = pd.DataFrame(topic_weights).fillna(0).values

# Keep the well-separated points (optional)
arr = arr[np.argmax(arr, axis=1) > 0.35]

# Dominant topic number in each doc
topic_num = np.argmax(arr, axis=1)

# t-SNE Dimension Reduction
# t-SNE Dimension Reduction
tsne_model = TSNE(n_components=2, verbose=1, random_state=0, angle=.99, init='pca',
    perplexity=2) # Set perplexity less than n_samples
tsne_lda = tsne_model.fit_transform(arr)
```



# Plot topic clusters using Bokeh

output\_notebook()

n\_topics = 7

mycolors = np.array([color for name, color in mcolors.TABLEAU\_COLORS.items()])

```
plot = figure(title="t-SNE Clustering of { } LDA Topics".format(n_topics), width=900,
height=700) # Changed plot_width to width, plot_height to height
plot.scatter(x=tsne_lda[:, 0], y=tsne_lda[:, 1], color=mycolors[topic_num])
show(plot)
```

# LDA visualization

pyLDavis.enable\_notebook()

```
vis = pyLDavis.gensim.prepare(lda_model, corpus, dictionary=lda_model.id2word)
vis
```

## OUTPUT :

```
# LDA visualization
pyLDavis.enable_notebook()
vis = pyLDavis.gensim.prepare(lda_model, corpus, dictionary=lda_model.id2word)
vis
```

<>:38: DeprecationWarning: invalid escape sequence '\s'  
<>:38: DeprecationWarning: invalid escape sequence '\s'  
<ipython-input-8-91f7d77e45d5>:38: DeprecationWarning: invalid escape sequence '\s'  
sent = re.sub('\s+', ' ', sent) # Remove newline characters  
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)  
Requirement already satisfied: spacy in /usr/local/lib/python3.11/dist-packages (3.7.5)  
Requirement already satisfied: gensim in /usr/local/lib/python3.11/dist-packages (4.3.3)  
Requirement already satisfied: wordcloud in /usr/local/lib/python3.11/dist-packages (1.9.4)  
Requirement already satisfied: pyLDavis in /usr/local/lib/python3.11/dist-packages (3.4.1)  
Requirement already satisfied: bokeh in /usr/local/lib/python3.11/dist-packages (3.6.3)  
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.1.8)  
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.4.2)  
Requirement already satisfied: regex<=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)  
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)  
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.11/dist-packages (from spacy) (3.0.12)  
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from spacy) (1.0.5)  
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.11/dist-packages (from spacy) (1.0.12)  
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy) (2.0.11)  
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy) (3.0.9)  
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in /usr/local/lib/python3.11/dist-packages (from spacy) (8.2.5)  
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.11/dist-packages (from spacy) (1.1.3)  
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.11/dist-packages (from spacy) (2.5.1)  
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.11/dist-packages (from spacy) (2.0.10)  
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in /usr/local/lib/python3.11/dist-packages (from spacy) (0.4.1)  
Requirement already satisfied: typer<1.0.0,>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from spacy) (0.15.2)  
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from spacy) (2.32.3)  
Requirement already satisfied: pydantic<1.8.1,>=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.11/dist-packages (from spacy) (2.10.6)  
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.11/dist-packages (from spacy) (3.1.6)  
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from spacy) (75.1.0)  
Requirement already satisfied: packaging<20.0 in /usr/local/lib/python3.11/dist-packages (from spacy) (24.2)  
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.11/dist-packages (from spacy) (3.5.0)  
Requirement already satisfied: nummv<=1.19.0 in /usr/local/lib/python3.11/dist-packages (from snacv) (1.76.4)

15s completed at 12:41 AM



```
[nltk_data] Downloading package stopwords to /root/nltk_data...  
[nltk_data] Package stopwords is already up-to-date!  
[(0,  
  '0.086*'service' + 0.086*'buy' + 0.086*'never' + 0.086*'delivery' + '  
  '0.086*'experience' + 0.086*'fast' + 0.086*'terrible' + 0.086*'great' + '  
  '0.086*'average' + 0.086*'special''),  
(1,  
  '0.216*'love' + 0.216*'amazing' + 0.216*'product' + 0.027*'bad' + '  
  '0.027*'well' + 0.027*'great' + 0.027*'quality' + 0.027*'special' + '  
  '0.027*'terrible' + 0.027*'average''),  
(2,  
  '0.063*'bad' + 0.063*'quality' + 0.063*'average' + 0.063*'special' + '  
  '0.063*'well' + 0.063*'product' + 0.063*'terrible' + 0.063*'fast' + '  
  '0.062*'buy' + 0.062*'amazing''),  
(3,  
  '0.063*'product' + 0.063*'bad' + 0.063*'quality' + 0.063*'well' + '  
  '0.063*'great' + 0.063*'special' + 0.063*'average' + 0.063*'fast' + '  
  '0.062*'buy' + 0.062*'amazing''),  
(4,  
  '0.063*'bad' + 0.063*'quality' + 0.063*'special' + 0.063*'amazing' + '  
  '0.063*'average' + 0.063*'experience' + 0.063*'well' + 0.062*'great' + '  
  '0.062*'delivery' + 0.062*'love''),  
(5,  
  '0.267*'well' + 0.267*'bad' + 0.033*'special' + 0.033*'quality' + '  
  '0.033*'amazing' + 0.033*'product' + 0.033*'great' + 0.033*'fast' + '  
  '0.033*'terrible' + 0.033*'love''),  
(6,  
  '0.063*'bad' + 0.063*'special' + 0.063*'quality' + 0.063*'average' + '  
  '0.063*'well' + 0.062*'product' + 0.062*'amazing' + 0.062*'great' + '
```

Topic 0

terrible buy  
great average fast  
delivery experience  
service never special

Topic 2

buy special terrible  
well  
product amazing bad  
average fast quality

Topic 4

experience amazing  
bad quality  
well delivery love  
special great average

Topic 1

average amazing  
well bad  
love product  
quality great special  
terrible

Topic 3

well  
buy average amazing  
quality bad special  
product fast great

Topic 5

love bad  
quality great  
product special fast  
well amazing  
terrible

```
[t-SNE] Computing 4 nearest neighbors...  
[t-SNE] Indexed 5 samples in 0.001s...  
[t-SNE] Computed neighbors for 5 samples in 0.001s...  
[t-SNE] Computed conditional probabilities for sample 5 / 5  
[t-SNE] Mean sigma: 0.127215  
[t-SNE] KL divergence after 250 iterations with early exaggeration: 71.987167  
[t-SNE] KL divergence after 1000 iterations: 0.053072
```

