Practical 07: Sentiment Analysis on IMDb Dataset

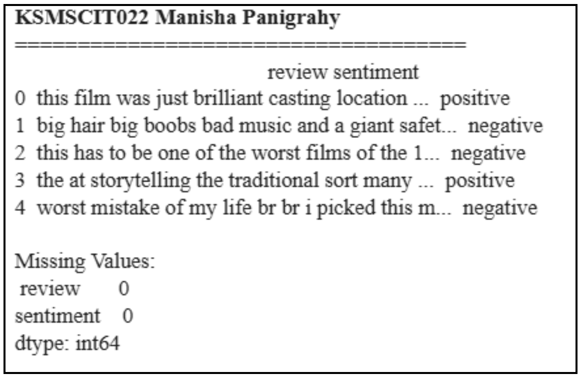
Aim: To perform sentiment analysis on the IMDb dataset by cleaning text data, analyzing sentiment distribution, and visualizing frequent words in positive and negative reviews.

Theory: Sentiment analysis is a natural language processing (NLP) technique used to determine the sentiment expressed in text data. The IMDb dataset consists of movie reviews labeled as positive or negative. The process involves text preprocessing, sentiment classification, and data visualization to gain insights into review patterns.

Code:

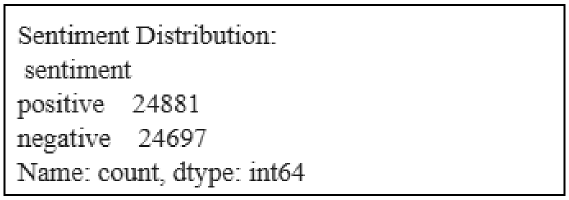
```
from keras.datasets import imdb
nltk.download('stopwords')
nltk.download('punkt')
stop_words = set(stopwords.words('english'))
def load_and_clean_data(num_words=10000):
    (train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words=num_words)
    word_index = imdb.get_word_index()
    reverse_word_index = {value: key for (key, value) in word_index.items()}
    def decode_review(encoded_review):
        return " ".join([reverse_word_index.get(i - 3, "?") for i in encoded_review if i >= 3])
    train_reviews = [decode_review(seq) for seq in train_data]
    test_reviews = [decode_review(seq) for seq in test_data]
    df_train = pd.DataFrame({
        "review": train_reviews,
        "sentiment": train_labels})
    df_test = pd.DataFrame({
        "review": test_reviews,
        "sentiment": test_labels})
    df = pd.concat([df_train, df_test], ignore_index=True)
    df["sentiment"] = df["sentiment"].map({0: "negative", 1: "positive"})
    df.drop_duplicates(inplace=True)
    df.dropna(inplace=True)
    if "review" not in df.columns or "sentiment" not in df.columns:
        raise ValueError("Dataset does not have required columns: 'review' and 'sentiment'")
    return df
df = load_and_clean_data(num_words=10000)
print(df.head())
print("\nMissing Values:\n", df.isnull().sum())
print("\nSentiment Distribution:\n", df["sentiment"].value_counts())
```

Program Output:



```
KSMSCIT022 Manisha Panigrahy
====================================
                                   review sentiment
0  this film was just brilliant casting location ...  positive
1  big hair big boobs bad music and a giant safet...  negative
2  this has to be one of the worst films of the 1...  negative
3  the at storytelling the traditional sort many ...  positive
4  worst mistake of my life br br i picked this m...  negative

Missing Values:
 review       0
sentiment    0
dtype: int64
```

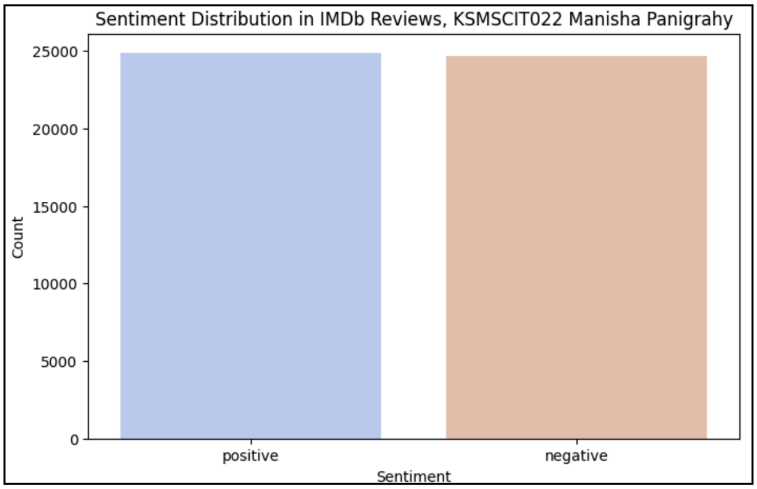


```
Sentiment Distribution:
 sentiment
positive    24881
negative    24697
Name: count, dtype: int64
```

Code:

```
print('\033[1mKSMSCIT022 Manisha Umesh Panigrahy\033[0m')
print('\033[1m=====================================\033[0m')
def clean_text(text):
    text = re.sub(r'<.*?>', '', text)  # Remove HTML tags
    text = re.sub(r'[^a-zA-Z ]', '', text)  # Remove non-alphabetic characters
    text = text.lower()
    words = text.split()
    words = [word for word in words if word not in stop_words]
    return ' '.join(words)
df['clean_review'] = df['review'].astype(str).apply(clean_text)
plt.figure(figsize=(8, 5))
sns.barplot(x=df['sentiment'].value_counts().index, y=df['sentiment'].value_counts().values, palette='coolwarm')
plt.title("Sentiment Distribution in IMDb Reviews, KSMSCIT022 Manisha Panigrahy")
plt.xlabel("Sentiment")plt.ylabel("Count")plt.show()
```

Program Output:

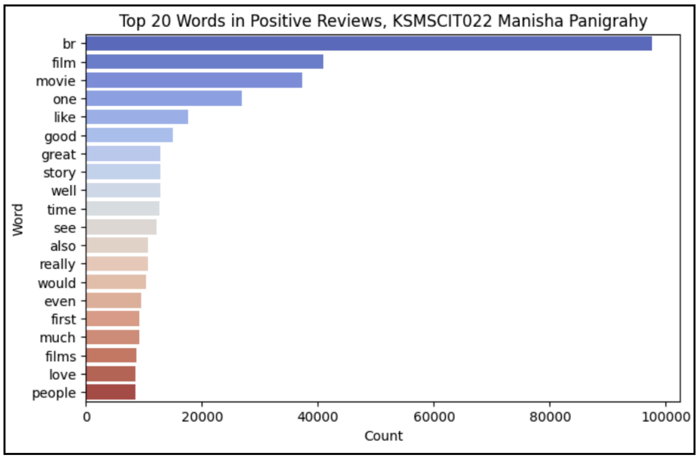


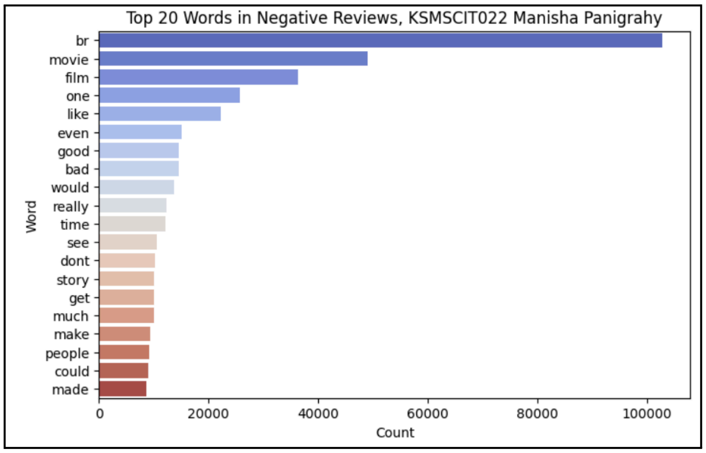
Code:

```
from collections import Counter
positive_words = ' '.join(df[df['sentiment'] == 'positive']['clean_review']).split()
negative_words = ' '.join(df[df['sentiment'] == 'negative']['clean_review']).split()
positive_word_counts = Counter(positive_words).most_common(20)
```

```
negative_word_counts = Counter(negative_words).most_common(20)
positive_df = pd.DataFrame(positive_word_counts, columns=['Word', 'Count'])
negative_df = pd.DataFrame(negative_word_counts, columns=['Word', 'Count'])
print("KSMSCIT007 Dhanraj Chinta")
plt.figure(figsize=(8, 5))
sns.barplot(x='Count', y='Word', data=positive_df, palette='coolwarm')
plt.title("Top 20 Words in Positive Reviews, KSMSCIT007 Dhanraj Chinta")
plt.show()
plt.figure(figsize=(8, 5))
sns.barplot(x='Count', y='Word', data=negative_df, palette='coolwarm')
plt.title("Top 20 Words in Negative Reviews, KSMSCIT007 Dhanraj Chinta")
plt.show()
```

Program Output:


Top 20 Words in Positive Reviews, KSMSCIT022 Manisha Panigrahy
Word
br
film
movie
one
like
good
great
story
well
time
see
also
really
would
even
first
much
films
love
people
Count


Top 20 Words in Negative Reviews, KSMSCIT022 Manisha Panigrahy
Word
br
movie
film
one
like
even
good
bad
would
really
time
see
dont
story
get
much
make
people
could
made
Count