NAME: KRISHNAKANT NANDKUMAR KOLNURE

PRN: 202401040087

DIVISION: CS3

BATCH: C31

ROLL NO: CS3-14

DATASET: Movie Review

Under The Guidance Of Course In-Charge,

Prof. Anjali Patil

1.Display first five rows of the data set

```
[1] import pandas as pd
    import numpy as np
```

Double-click (or enter) to edit

```
[3] df = pd.read_csv('/content/sample_data/IMDB Dataset.csv')
```

```
df.head()
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

Next steps: Generate code with df | View recommended plots | New interactive sheet

2. Find total number of reviews

3. Check for missing values

4. Number of positive and negative reviews

```
[6] print("Total reviews:", df.shape[0])
```

```
Total reviews: 50000
```

```
print("Missing values:\n", df.isnull().sum())
```

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

```
[8] print(df['sentiment'].value_counts())
```

```
sentiment
positive    25000
negative    25000
Name: count, dtype: int64
```

5. Percentage of positive and negative reviews

6. Find length of each review

7. Add new column for word count

```
[10] print((df['sentiment'].value_counts(normalize=True) * 100).round(2))

sentiment
positive    50.0
negative    50.0
Name: proportion, dtype: float64

[30] df['review_length'] = df['review'].apply(len)
     print(df[['review', 'review_length']].head())

                                              review  review_length
0  One of the other reviewers has mentioned that ...           1761
1  A wonderful little production. <br /><br />The...            998
2  I thought this was a wonderful way to spend ti...            926
3  Basically there's a family where a little boy ...            748
4  Petter Mattei's "Love in the Time of Money" is...           1317

df['word_count'] = df['review'].apply(lambda x: len(x.split()))
print(df[['review', 'word_count']].head())

                                              review  word_count
0  One of the other reviewers has mentioned that ...         307
1  A wonderful little production. <br /><br />The...         162
2  I thought this was a wonderful way to spend ti...         166
3  Basically there's a family where a little boy ...         138
4  Petter Mattei's "Love in the Time of Money" is...         230
```

8. Review with maximum number of words

9. Review with minimum number of words

10. Average number of words per review

```
[14] max_word_review = df.loc[df['word_count'].idxmax()]
     print("Review with most words:\n", max_word_review['review'])

Review with most words:
 Match 1: Tag Team Table Match Bubba Ray and Spike Dudley vs Eddie Guerrero and Chris Benoit Bubba Ray and Spike Dudley started things off with a Tag Team Table Match against Eddie Gue

min_word_review = df.loc[df['word_count'].idxmin()]
print("Review with fewest words:\n", min_word_review['review'])

Review with fewest words:
 Primary plot!Primary direction!Poor interpretation.

[17] print("Average word count:", df['word_count'].mean())

Average word count: 231.15694
```

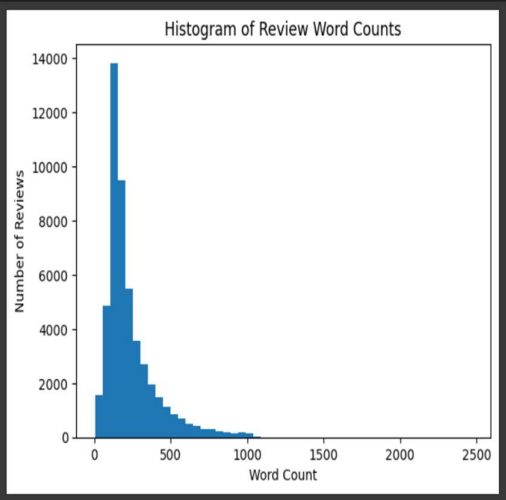
11. Standard deviation of review lengths

```
print("Std deviation of review lengths:", np.std(df['review_length']))
```

```
Std deviation of review lengths: 989.7181170827207
```

12. Plot histogram of word counts

```
import matplotlib.pyplot as plt
plt.hist(df['word_count'], bins=50)
plt.xlabel('Word Count')
plt.ylabel('Number of Reviews')
plt.title('Histogram of Review Word Counts')
plt.show()
```



13. Top 10 most frequent words

```
from collections import Counter
all_words = ' '.join(df['review']).lower().split()
word_freq = Counter(all_words)
print("Top 10 words:", word_freq.most_common(10))
```

```
Top 10 words: [('the', 638861), ('a', 316615), ('and', 313637), ('of', 286661), ('to', 264573), ('is', 204876), ('in', 179807), ('i', 141587), ('this', 138483), ('that', 130140)]
```

14. Average review length for positive and negative reviews

15. Number of reviews containing the word "good"

16. Number of reviews containing the word "bad"

```
[21] print(df.groupby('sentiment')['word_count'].mean())

sentiment
negative    229.46456
positive    232.84932
Name: word_count, dtype: float64

good_reviews = df['review'].str.contains('good', case=False).sum()
print("Reviews mentioning 'good':", good_reviews)

Reviews mentioning 'good': 19472

[23] bad_reviews = df['review'].str.contains('bad', case=False).sum()
print("Reviews mentioning 'bad':", bad_reviews)

Reviews mentioning 'bad': 12704
```

17. New column flagging "excellent"

18. Proportion of positive reviews mentioning "excellent"

19. Remove duplicate reviews

```
[24] df['mentions_excellent'] = df['review'].str.contains('excellent', case=False)
     print(df[['review', 'mentions_excellent']].head())

                                                 review  mentions_excellent
0  One of the other reviewers has mentioned that ...               False
1  A wonderful little production. <br /><br />The...               False
2  I thought this was a wonderful way to spend ti...               False
3  Basically there's a family where a little boy ...               False
4  Petter Mattei's "Love in the Time of Money" is...               False

     positive_reviews = df[df['sentiment'] == 'positive']
     excellent_in_positive = positive_reviews['mentions_excellent'].mean()
     print("Proportion of positive reviews mentioning 'excellent':", excellent_in_positive)

Proportion of positive reviews mentioning 'excellent': 0.11744

[26] df = df.drop_duplicates(subset='review')
     print("Shape after removing duplicates:", df.shape)

Shape after removing duplicates: (49582, 5)
```

20. Save cleaned dataset

```
[29] df.to_csv('/content/sample_data/IMDB Dataset.csv', index=False)
     print("Cleaned dataset saved.")

Cleaned dataset saved.
```

THANK YOU