1.Upload the CSV file

```
from google.colab import files
uploaded = files.upload()
```

2.Read the file into a pandas data frame.

```
import pandas as pd

df = pd.read_csv("Womens Clothing E-Commerce Reviews.csv")
```

3.Display the first 10 rows

```
df.head(10) # displays first 10 rows
```

4.Import the libraries

```
import numpy as np

import nltk

nltk.download('vader_lexicon') # download the VADER lexicon for sentiment analysis

from nltk.sentiment.vader import Sentiment IntensityAnalyzer

# create a sentiment analyzer object

sid = SentimentIntensityAnalyzer()
```

5.Calculate sentiment scores

```
[5] # replace missing values with empty strings

df['Review Text'] = df['Review Text').replace(np.nan, regex=True)

# iterate over the review text column and calculate the sentiment scores

sentiment_scores = []

for text in df['Review Text']:

scores = sid.polarity_scores (text)

sentiment_scores.append(scores['compound'])

# add the sentiment scores as a new column in the DataFrame

df['Sentiment Score'] = sentiment_scores
```

6.Define a function

```
# define a function to map the compound scores to sentiment labels

def get_sentiment_label(score):

if score >= 0.05:

return 'Positive'

elif score <= -0.05:

return 'Negative'

else:

return 'Neutral'

# apply the get_sentiment_label function to the sentiment scores to get the sentiment labels
sentiment_labels = df['Sentiment Score'].apply(get_sentiment_label)

#add the sentiment labels as a new column in the DataFrame df['Sentiment Label'] =
sentiment_labels
```

7.Display the first 10 rows with ‘’Sentiment Score”

```
df.head(10) # displays first 10 rows with newly added columns “Sentiment Score”
```

Data Visualization

1.Creating a pie chart

```
import matplotlib.pyplot as plt

#count the number of reviews for each sentiment-label

sentiment_counts = df['Sentiment Label'].value_counts()

#create a pie chart

plt.pie(sentiment_counts, labels sentiment_counts.index, autopct='%1.1f%%')

plt.title('Sentiment Distribution')

plt.show()
```

2.Sentiment is distributed for each rating

```
[9] import matplotlib.pyplot as plt

# group the data by rating and sentiment label, and count the number of reviews in each
group grouped = df.groupby(['Rating', 'Sentiment Label']).size().reset_index(name='Count')

# iterate over each rating and plot a pie chart of the sentiment label distribution

for rating in range(1, 6):

data = grouped [grouped [Rating'] == rating]

plt.pie(data['Count'], labels=data['Sentiment Label'], autopct='%1.1f%%')

plt.title(f'Sentiment Label Distribution for Rating {rating}')

plt.show()
```

3.Correlation between sentiment and rating scores.

```
import matplotlib.pyplot as plt

# Plot the graph

fig = plt.figure(figsize=(8,6))

df.groupby(['Rating', 'Sentiment Label']).size().unstack().plot(kind='bar', stacked=True)

plt.title('Sentiment Label vs Rating')

plt.xlabel('Rating')

plt.ylabel('Count')

plt.show()
```