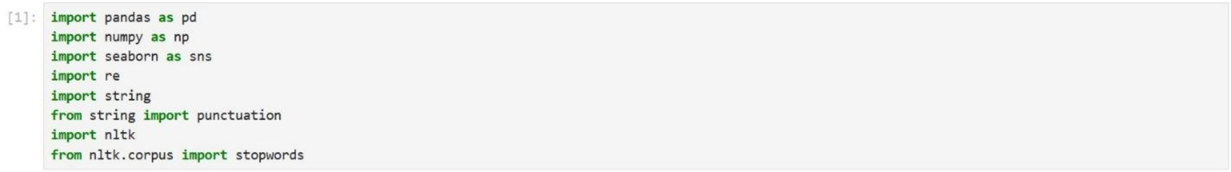
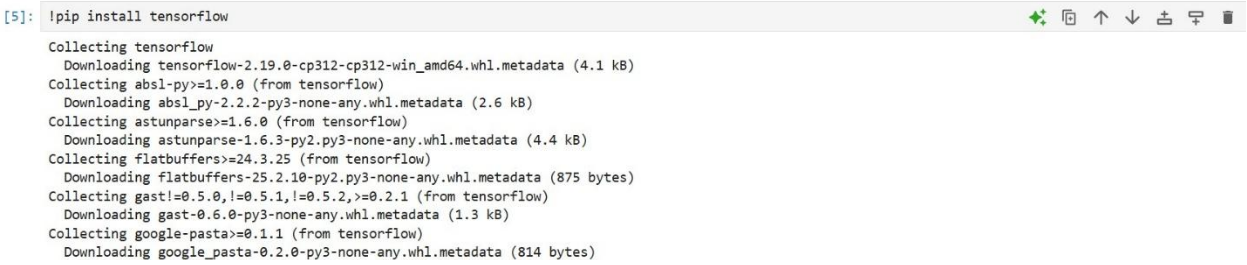
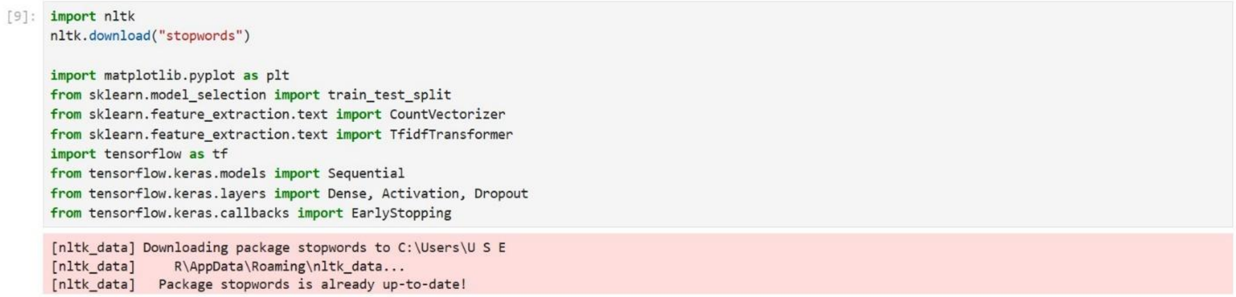
Lab Exercise 2	MUHAMMAD FAIZ BIN ABD RAZAK	NBSC2405A	ICT 550	2023911655



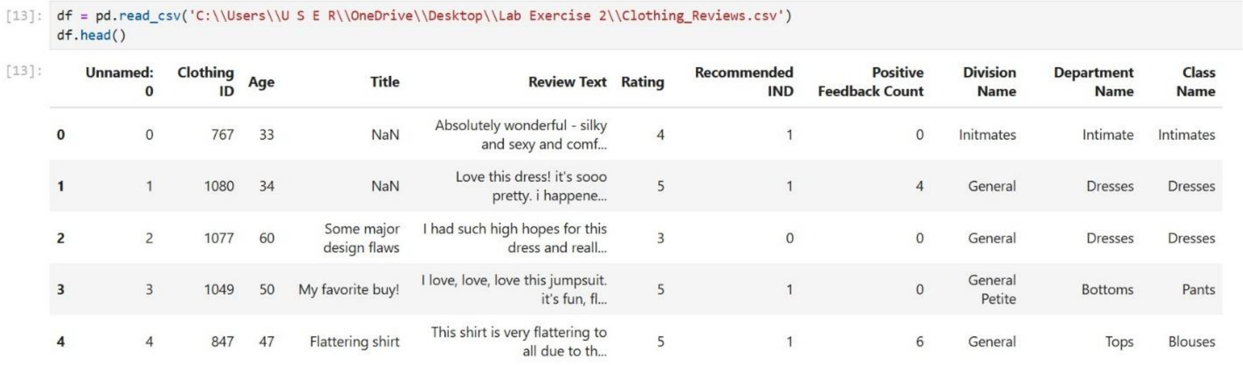
```
[1]: import pandas as pd
     import numpy as np
     import seaborn as sns
     import re
     import string
     from string import punctuation
     import nltk
     from nltk.corpus import stopwords
```



```
[5]: !pip install tensorflow

     Collecting tensorflow
       Downloading tensorflow-2.19.0-cp312-cp312-win_amd64.whl.metadata (4.1 kB)
     Collecting absl-py>=1.0.0 (from tensorflow)
       Downloading absl_py-2.2.2-py3-none-any.whl.metadata (2.6 kB)
     Collecting astunparse>=1.6.0 (from tensorflow)
       Downloading astunparse-1.6.3-py2.py3-none-any.whl.metadata (4.4 kB)
     Collecting flatbuffers>=24.3.25 (from tensorflow)
       Downloading flatbuffers-25.2.10-py2.py3-none-any.whl.metadata (875 bytes)
     Collecting gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 (from tensorflow)
       Downloading gast-0.6.0-py3-none-any.whl.metadata (1.3 kB)
     Collecting google-pasta>=0.1.1 (from tensorflow)
       Downloading google_pasta-0.2.0-py3-none-any.whl.metadata (814 bytes)
```



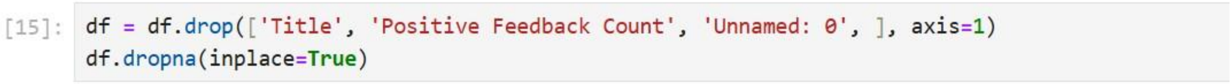
```
[9]: import nltk
     nltk.download("stopwords")

     import matplotlib.pyplot as plt
     from sklearn.model_selection import train_test_split
     from sklearn.feature_extraction.text import CountVectorizer
     from sklearn.feature_extraction.text import TfidfTransformer
     import tensorflow as tf
     from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Dense, Activation, Dropout
     from tensorflow.keras.callbacks import EarlyStopping

     [nltk_data] Downloading package stopwords to C:\Users\U S E
     [nltk_data]     R\AppData\Roaming\nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!
```



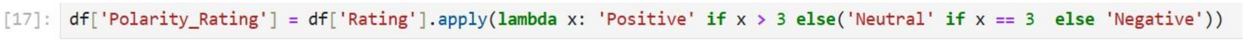
```
[13]: df = pd.read_csv('C:\\Users\\U S E R\\OneDrive\\Desktop\\Lab Exercise 2\\Clothing_Reviews.csv')
      df.head()
```

[13]:

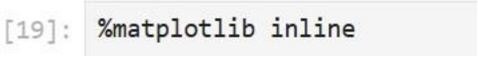
	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	Tops	Blouses



```
[15]: df = df.drop(['Title', 'Positive Feedback Count', 'Unnamed: 0', ], axis=1)
      df.dropna(inplace=True)
```



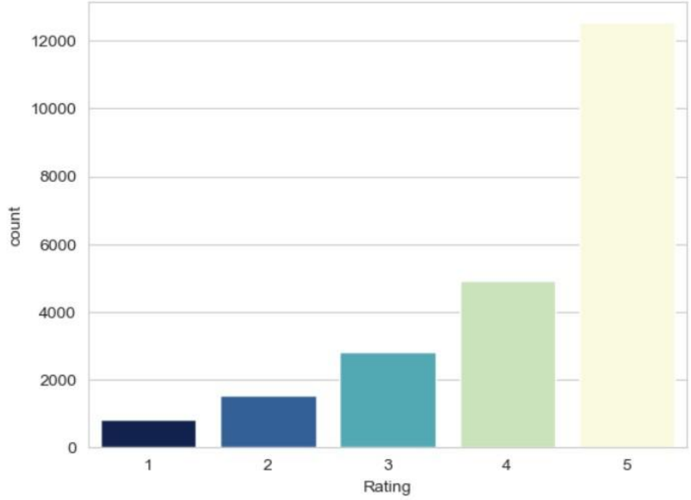
```
[17]: df['Polarity_Rating'] = df['Rating'].apply(lambda x: 'Positive' if x > 3 else('Neutral' if x == 3  else 'Negative'))
```



```
[19]: %matplotlib inline
```

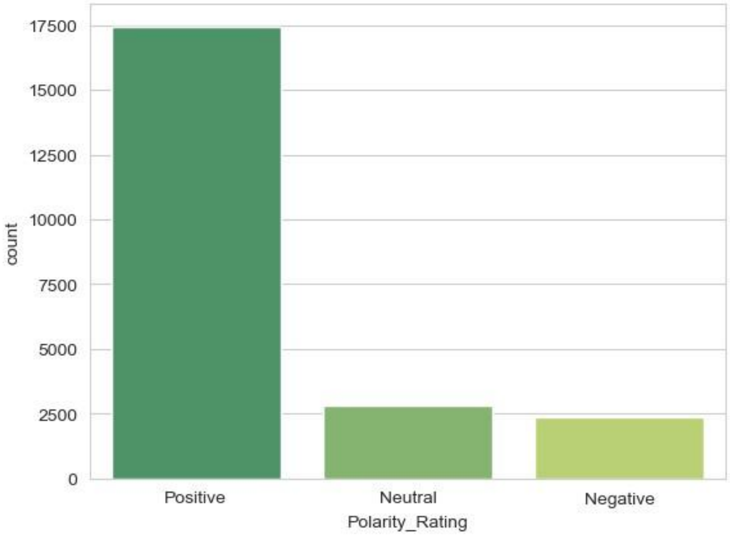
[41]:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_style('whitegrid')
ax = sns.countplot(x='Rating', data=df, hue='Rating', palette='YlGnBu_r')
ax.legend_.remove()  # Hides the legend
plt.show()
```



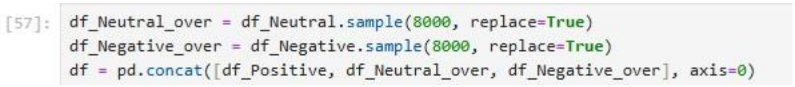
[53]:

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set_style('whitegrid')
ax = sns.countplot(x='Polarity_Rating', data=df, hue='Polarity_Rating', palette='summer')

# Only remove the legend if it exists
if ax.get_legend() is not None:
    ax.legend_.remove()

plt.show()
```

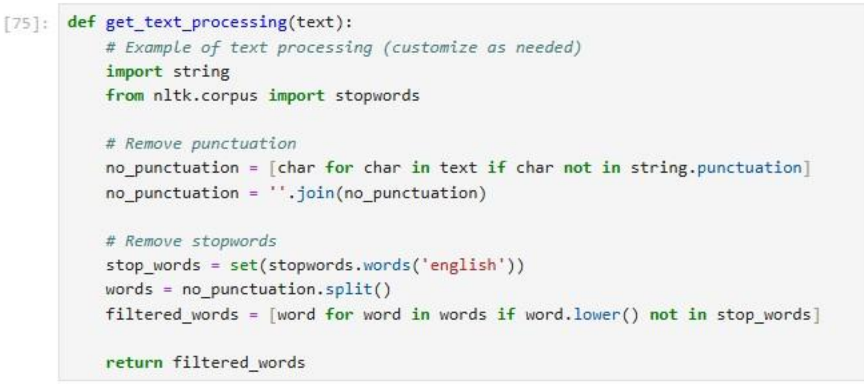


[55]:

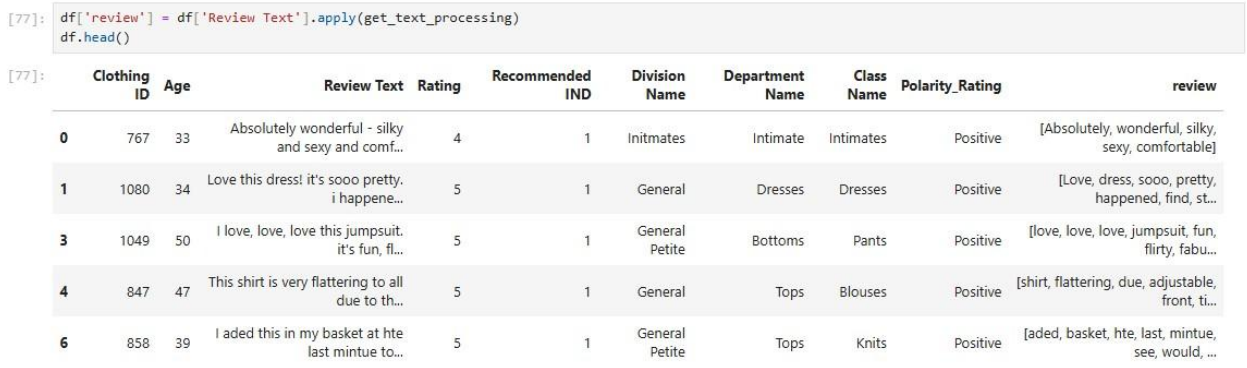
```
df_Positive = df[df['Polarity_Rating'] == 'Positive'][0:8000]
df_Neutral = df[df['Polarity_Rating'] == 'Neutral']
df_Negative = df[df['Polarity_Rating'] == 'Negative']
```



```
[57]: df_Neutral_over = df_Neutral.sample(8000, replace=True)
      df_Negative_over = df_Negative.sample(8000, replace=True)
      df = pd.concat([df_Positive, df_Neutral_over, df_Negative_over], axis=0)
```



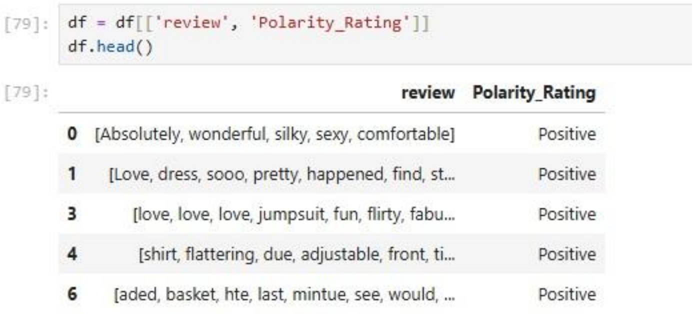
```
[75]: def get_text_processing(text):
          # Example of text processing (customize as needed)
          import string
          from nltk.corpus import stopwords

          # Remove punctuation
          no_punctuation = [char for char in text if char not in string.punctuation]
          no_punctuation = ''.join(no_punctuation)

          # Remove stopwords
          stop_words = set(stopwords.words('english'))
          words = no_punctuation.split()
          filtered_words = [word for word in words if word.lower() not in stop_words]

          return filtered_words
```



```
[77]: df['review'] = df['Review Text'].apply(get_text_processing)
      df.head()
```

[77]:

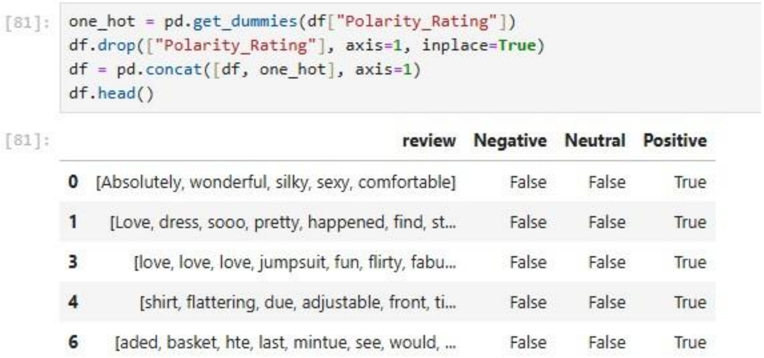
	Clothing ID	Age	Review Text	Rating	Recommended IND	Division Name	Department Name	Class Name	Polarity_Rating	review
0	767	33	Absolutely wonderful - silky and sexy and comf...	4	1	Initmates	Intimate	Intimates	Positive	[Absolutely, wonderful, silky, sexy, comfortable]
1	1080	34	Love this dress! it's sooo pretty. i happene...	5	1	General	Dresses	Dresses	Positive	[Love, dress, sooo, pretty, happened, find, st...
3	1049	50	I love, love, love this jumpsuit. it's fun, fl...	5	1	General Petite	Bottoms	Pants	Positive	[love, love, love, jumpsuit, fun, flirty, fabu...
4	847	47	This shirt is very flattering to all due to th...	5	1	General	Tops	Blouses	Positive	[shirt, flattering, due, adjustable, front, ti...
6	858	39	I aded this in my basket at hte last mintue to...	5	1	General Petite	Tops	Knits	Positive	[aded, basket, hte, last, mintue, see, would, ...



```
[79]: df = df[['review', 'Polarity_Rating']]
      df.head()
```

[79]:

	review	Polarity_Rating
0	[Absolutely, wonderful, silky, sexy, comfortable]	Positive
1	[Love, dress, sooo, pretty, happened, find, st...	Positive
3	[love, love, love, jumpsuit, fun, flirty, fabu...	Positive
4	[shirt, flattering, due, adjustable, front, ti...	Positive
6	[aded, basket, hte, last, mintue, see, would, ...	Positive



```
[81]: one_hot = pd.get_dummies(df["Polarity_Rating"])
      df.drop(["Polarity_Rating"], axis=1, inplace=True)
      df = pd.concat([df, one_hot], axis=1)
      df.head()
```

[81]:

	review	Negative	Neutral	Positive
0	[Absolutely, wonderful, silky, sexy, comfortable]	False	False	True
1	[Love, dress, sooo, pretty, happened, find, st...	False	False	True
3	[love, love, love, jumpsuit, fun, flirty, fabu...	False	False	True
4	[shirt, flattering, due, adjustable, front, ti...	False	False	True
6	[aded, basket, hte, last, mintue, see, would, ...	False	False	True

[91]:
```
!pip install afinn
```

```
Collecting afinn
  Downloading afinn-0.1.tar.gz (52 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: afinn
  Building wheel for afinn (setup.py): started
  Building wheel for afinn (setup.py): finished with status 'done'
  Created wheel for afinn: filename=afinn-0.1-py3-none-any.whl size=53438 sha256=ace26e0af6eda4d1e66d9a672b2427f3ba6f428f6c5186ba5277543ff0e197c9
  Stored in directory: c:\users\u s e r\appdata\local\pip\cache\wheels\f9\72\27\74994e77200dae3d6aea2b546264500cee21f738c51241320b
Successfully built afinn
Installing collected packages: afinn
Successfully installed afinn-0.1
```

[93]:
```
#lexicon-based sentiment analysis
from afinn import Afinn
import pandas as pd

#instantiate afinn
afn = Afinn()

#creating list sentences
news_df = [
    'data analytics is a great stuff',
    'i hate flowers',
    'hes kind and smart',
    'we are kind to good people'
]


# compute scores (polarity) and labels
scores = [afn.score(article) for article in news_df]
sentiment = ['positive' if score > 0
else 'negative' if score < 0
else 'neutral'
for score in scores]

# dataframe creation
df = pd.DataFrame()
df['topic'] =  news_df
df['scores'] = scores
df['sentiments'] = sentiment
print(df)
```

```
                             topic  scores sentiments
0  data analytics is a great stuff     3.0   positive
1                   i hate flowers    -3.0   negative
2               hes kind and smart     3.0   positive
3       we are kind to good people     5.0   positive
```