

Amazon Reviews Sentiment Analysis

In []:

```
import numpy as np # Linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LinearRegression
from keras.preprocessing.sequence import pad_sequences
```

In [6]:

```
from collections import Counter
import nltk
import seaborn as sns
import string
from nltk.corpus import stopwords
# import re
# from autocorrect import spell
import regex as re
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

from keras.models import Sequential
from keras.layers import Dense
from keras.backend import eval
from keras.optimizers import Adam
from keras.layers import LSTM
from tensorflow.keras.layers import Embedding
from keras.layers import Dropout
from keras.layers.convolutional import Conv1D, MaxPooling1D
```

In [8]:

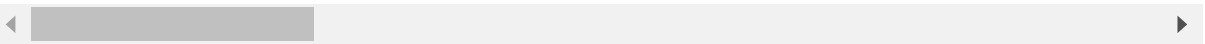
```
data=pd.read_csv('1429_1.csv')
data.head()
```

C:\Users\Warren\AppData\Local\Temp\ipykernel_3552\2787805710.py:1: DtypeWarning: Columns (1,10) have mixed types. Specify dtype option on import or set low_memory=False.
data=pd.read_csv('1429_1.csv')

Out[8]:

	id	name	asins	brand	categories	
0	AVqklhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/530044€
1	AVqklhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/530044€
2	AVqklhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/530044€
3	AVqklhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/530044€
4	AVqklhwDv8e3D1O-lebb	All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi,...	B01AHB9CN2	Amazon	Electronics,iPad & Tablets,All Tablets,Fire Ta...	841667104676,amazon/530044€

5 rows × 21 columns



In [9]:

```
review=pd.DataFrame(data.groupby('reviews.rating').size().sort_values(ascending=False).rename  
review.head()
```

Out[9]:

	reviews.rating	No of Users
0	5.0	23775
1	4.0	8541
2	3.0	1499
3	1.0	410
4	2.0	402

In [10]:

```
permanent = data[['reviews.rating' , 'reviews.text' , 'reviews.title' , 'reviews.username']]  
mpermanent=permanent.dropna()  
mpermanent.head()
```

Out[10]:

	reviews.rating	reviews.text	reviews.title	reviews.username
0	5.0	This product so far has not disappointed. My c...	Kindle	Adapter
1	5.0	great for beginner or experienced person. Boug...	very fast	truman
2	5.0	Inexpensive tablet for him to use and learn on...	Beginner tablet for our 9 year old son.	DaveZ
3	4.0	I've had my Fire HD 8 two weeks now and I love...	Good!!!	Shacks
4	5.0	I bought this for my grand daughter when she c...	Fantastic Tablet for kids	explore42

In [11]:

```
check = mpermanent[mpermanent["reviews.text"].isnull()]  
check.head()
```

Out[11]:

reviews.rating	reviews.text	reviews.title	reviews.username
----------------	--------------	---------------	------------------

In [12]:

```
actualrating = mpermanent[(mpermanent['reviews.rating'] == 1) | (mpermanent['reviews.rating']  
actualrating.shape
```

Out[12]:

(24178, 4)

In [13]:

```
y = actualrating['reviews.rating']
x = actualrating['reviews.text'].reset_index()
```

In [14]:

```
len(y)
```

Out[14]:

24178

In [15]:

```
X = x['reviews.text']
print(X)
```

```
0      This product so far has not disappointed. My c...
1      great for beginner or experienced person. Boug...
2      Inexpensive tablet for him to use and learn on...
3      I bought this for my grand daughter when she c...
4      This amazon fire 8 inch tablet is the perfect ...
      ...
24173   This is exactly like any other usb power charg...
24174   Amazon should include this charger with the Ki...
24175   Love my Kindle Fire but I am really disappoint...
24176   I was surprised to find it did not come with a...
24177   to spite the fact that i have nothing but good...
Name: reviews.text, Length: 24178, dtype: object
```

In [16]:

```
print(len(X))
```

24178

In [17]:

```
import string
from nltk.corpus import stopwords
# stop=set(stopwords.words('english'))
def text_process(text):
    """
    Takes in a string of text, then performs the following:
    1. Remove all punctuation
    2. Remove all stopwords
    3. Return the cleaned text as a list of words
    """
    nopunc = [char for char in text if char not in string.punctuation]
    nopunc = ''.join(nopunc)

    return [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]
```

In [18]:

```
tokens = X[0].split()
print(tokens)
```

```
['This', 'product', 'so', 'far', 'has', 'not', 'disappointed.', 'My', 'children', 'love', 'to', 'use', 'it', 'and', 'I', 'like', 'the', 'ability', 'to', 'monitor', 'control', 'what', 'content', 'they', 'see', 'with', 'ease.']
```

In [19]:

```
sample_text = "Hey there! This is a sample review, which happens to contain punctuations."
print(text_process(sample_text))
```

```
['Hey', 'sample', 'review', 'happens', 'contain', 'punctuations']
```

In [20]:

```
from sklearn.feature_extraction.text import CountVectorizer
# next we need to vectorize our input variable (X)
# we use the count vectoriser function and the analyser we use is the above lines of code
# this should return a vector array
bow_transformer = CountVectorizer(analyzer=text_process).fit(X)
```

In [21]:

```
len(bow_transformer.vocabulary_)
```

Out[21]:

```
16954
```

In [22]:

```
review_24 = X[24]
```

In [23]:

```
bow_25 = bow_transformer.transform([review_24])
bow_25
```

Out[23]:

```
<1x16954 sparse matrix of type '<class 'numpy.int64'>'
      with 14 stored elements in Compressed Sparse Row format>
```

In [24]:

```
print(bow_25)
```

```
(0, 1816)      1
(0, 2476)      1
(0, 5664)      1
(0, 7835)      1
(0, 10671)     1
(0, 10778)     1
(0, 11499)     1
(0, 11691)     1
(0, 11940)     1
(0, 14868)     1
(0, 15195)     1
(0, 15507)     1
(0, 16158)     1
(0, 16540)     1
```

In [25]:

```
X = bow_transformer.transform(X)
```

In [26]:

```
#Lets start training the model
from sklearn.model_selection import train_test_split
#using 30% of the data for testing, this will be revised once we do not get the desired accuracy
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

In [27]:

```
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
nb.fit(X_train, y_train)
```

Out[27]:

```
▼ MultinomialNB
MultinomialNB()
```

In [28]:

```
preds = nb.predict(X_test)
```

In [29]:

```
from sklearn.metrics import confusion_matrix, classification_report
print(confusion_matrix(y_test, preds))
print('\n')
print(classification_report(y_test, preds))
nb.score(X_train, y_train)
```

```
[[ 13 104]
 [ 52 7085]]
```

	precision	recall	f1-score	support
1.0	0.20	0.11	0.14	117
5.0	0.99	0.99	0.99	7137
accuracy			0.98	7254
macro avg	0.59	0.55	0.57	7254
weighted avg	0.97	0.98	0.98	7254

Out[29]:

0.9839872370597967

In [30]:

```
from sklearn.svm import SVC
clf = SVC()
clf.fit(X_train, y_train)
predsvm=clf.predict(X_test)
```

In [31]:

```
from sklearn.metrics import confusion_matrix, classification_report
print(confusion_matrix(y_test, preds))
print('\n')
print(classification_report(y_test, predsvm))
predsvm=clf.predict(X_test)
clf.score(X_train,y_train)
```

```
[[ 13 104]
 [ 52 7085]]
```

	precision	recall	f1-score	support
1.0	0.00	0.00	0.00	117
5.0	0.98	1.00	0.99	7137
accuracy			0.98	7254
macro avg	0.49	0.50	0.50	7254
weighted avg	0.97	0.98	0.98	7254

C:\Users\Warren\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))

C:\Users\Warren\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))

C:\Users\Warren\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))

Out[31]:

0.9908414086504372

In [32]:

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X, y)
```

Out[32]:

```
▼ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```


In [33]:

```
predsknn=neigh.predict(X_test)
```

In [34]:

```
from sklearn.metrics import confusion_matrix, classification_report
print(confusion_matrix(y_test, predsknn))
print('\n')
print(classification_report(y_test, predsknn))
neigh.score(X_train,y_train)
```

```
[[ 12 105]
 [  0 7137]]
```

	precision	recall	f1-score	support
1.0	1.00	0.10	0.19	117
5.0	0.99	1.00	0.99	7137
accuracy			0.99	7254
macro avg	0.99	0.55	0.59	7254
weighted avg	0.99	0.99	0.98	7254

Out[34]:

```
0.9835736232569132
```

In [35]:

```
from sklearn.ensemble import GradientBoostingClassifier
model= GradientBoostingClassifier(n_estimators=100, learning_rate=1.0, max_depth=1, random_state=0)
model.fit(X_train, y_train)
predicted= model.predict(X_test)
```

In [36]:

```
from sklearn.metrics import confusion_matrix, classification_report
print(confusion_matrix(y_test, predicted))
print('\n')
print(classification_report(y_test, predicted))
model.score(X_train,y_train)
```

```
[[ 0 117]
 [ 0 7137]]
```

	precision	recall	f1-score	support
1.0	0.00	0.00	0.00	117
5.0	0.98	1.00	0.99	7137
accuracy			0.98	7254
macro avg	0.49	0.50	0.50	7254
weighted avg	0.97	0.98	0.98	7254

C:\Users\Warren\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

C:\Users\Warren\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

C:\Users\Warren\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

```
_warn_prf(average, modifier, msg_start, len(result))
```

Out[36]:

0.9826873079650201

In [37]:

```
positive_review = actualrating['reviews.text'][2]
positive_review
```

Out[37]:

'Inexpensive tablet for him to use and learn on, step up from the NABI. He was thrilled with it, learn how to Skype on it already...'

In [38]:

```
positive_review_transformed = bow_transformer.transform([positive_review])
nb.predict(positive_review_transformed)[0]
```

Out[38]:

5.0

In [39]:

```
positive_review = actualrating['reviews.text'][11]
positive_review
```

Out[39]:

'Excellent product. Easy to use, large screen makes watching movies and reading easier.'

In [40]:

```
positive_review_transformed = bow_transformer.transform([positive_review])
model.predict(positive_review_transformed)[0]
```

Out[40]:

5.0

In [41]:

```
negative_review = mpermanent['reviews.text'][34650]
print(negative_review)
```

Although the description states that this will work on my Kindle Fire HDX 8.9, when I plug the charger to the Kindle, I receive a message that the charger is not of sufficient wattage to properly charge the Kindle. I bought this charger as a back up for when I travel. The original charger which came with the Kindle does not show this warning when I plug it in.

In [42]:

```
negative_review_transformed = bow_transformer.transform([negative_review])
nb.predict(negative_review_transformed)[0]
```

Out[42]:

1.0

In [43]:

```
negative_review_transformed = bow_transformer.transform([negative_review])
neigh.predict(negative_review_transformed)[0]
```

Out[43]:

5.0

In [44]:

```
negative_review = mpermanent['reviews.text'][34656]  
print(negative_review)
```

Amazon should include this charger with the Kindle. The fact that they're charging us extra for something that should be included is a sign of cheapness. Plus, you can use any micro-USB phone charger instead of this to charge your Kindle. Save your money.

In [45]:

```
negative_review_transformed = bow_transformer.transform([negative_review])  
nb.predict(negative_review_transformed)[0]
```

Out[45]:

1.0

In [46]:

```
negative_review_transformed = bow_transformer.transform([negative_review])  
neigh.predict(negative_review_transformed)[0]
```

Out[46]:

1.0

In []: