SENTIMENT ANALYSIS ON AMAZON KINDLE REVIEWS

HEMANT VG

11910284

INTEGRATED B.TECH-M.TECH

LOVELY PROFESSIONAL UNIVERSITY

DECALARATION

I HEMANT VG declare that the project done on Sentiment analysis on Amazon Kindle Reviews is done by me. With the availability of source code as well as the report on my github repository

Github link: https://github.com/Hemantvg/SENTIMENT-ANALYSIS-ON-AMAZON-KINDLE-REVIEW

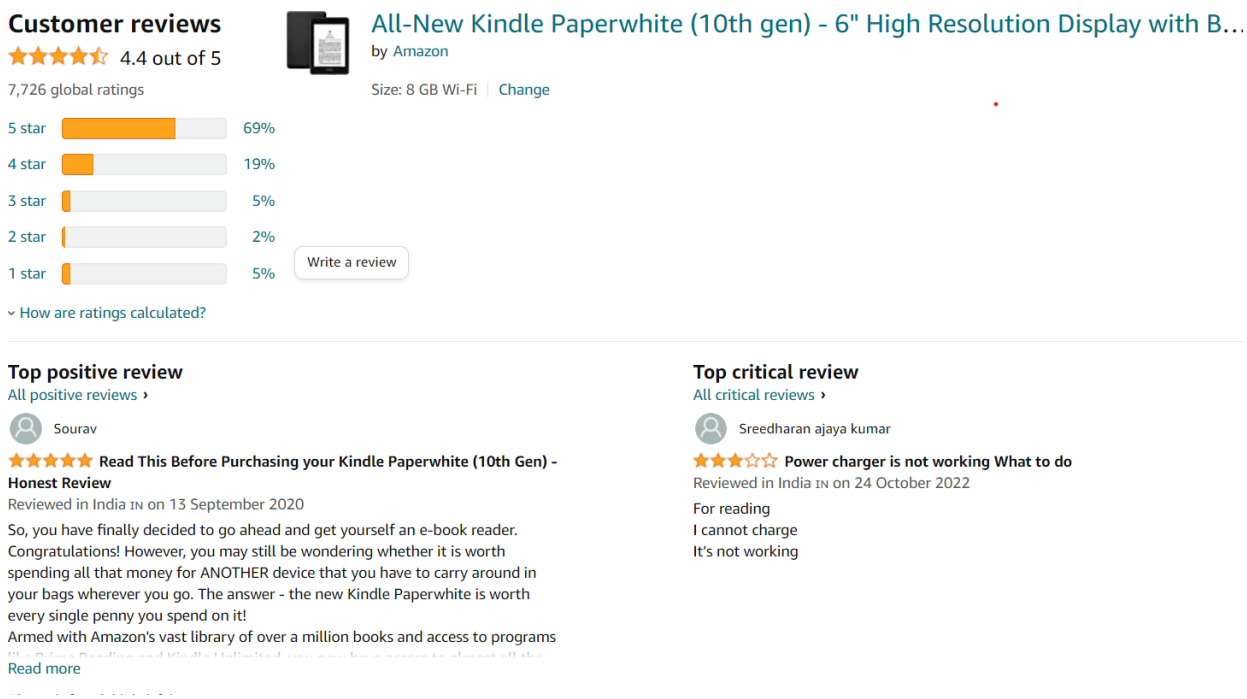
BONAFIDE CERTIFICATE

Certified That This Project Report **“Sentiment Analysis on Amazon Kindle Reviews”** is the bonafide work of **“HEMANT VG”** who carried out the project work under supervision.

TABLE OF CONTENTS

INTRODUCTION:

Sentiment Analysis on Amazon Kindle Review, So Generally What is a Sentiment Analysis, Sentiment analysis is a method for recognising the emotions connected to the text in natural language processing. Monitoring consumer comments on social media, monitoring for brands and campaigns are typical uses for sentiment analysis. This is a common method used by businesses to identify and group ideas regarding a certain good, service, or concept. Text is mined for sentiment and subjective information using data mining, machine learning, and artificial intelligence (AI). Systems for sentiment analysis assist businesses in extracting information from unstructured, disorganised language found in online sources including emails, blog posts, support tickets, web chats, social media channels, forums, and comments. Algorithms use rule-based, automated, or hybrid techniques to replace manual data processing. While automated systems use machine learning to learn from data, rule-based systems execute sentiment analysis using predetermined, lexicon-based rules. Combining the two methods results in a hybrid sentiment analysis. Opinion mining may extract the subject, opinion holder, and the degree of positivity and negative from text in addition to recognising sentiment. Additionally, other scopes, including document, paragraph, sentence, and sub-sentence levels, can be used for sentiment analysis. Brandwatch, Hootsuite, Lexalytics, NetBase, Sprout Social, Sysomos, and Zoho are some of the vendors that provide platforms or SaaS tools for sentiment analysis. Utilizing these technologies allows businesses to evaluate client input more often and react proactively to market shifts in opinion. The Certain usage the Sentiment Analysis on a certain product or reviewing or organization, how it is helpful to them. More people, companies, organisations, and governments are leveraging information from social media to guide their actions. When looking to purchase a product, a person no longer just asks those in his immediate vicinity but also searches the Internet for reviews, debates, and other information. An organisation may also use the Internet to evaluate the quality of its goods and services. In a similar vein, it is simple for governments to learn about significant occurrences in other nations as well as to receive public opinion on their policies. However, because of the growth of different websites, it is still challenging to find and monitor opinion websites online and to extract information from them. The typical human reader will struggle to locate pertinent websites, extract remarks from them, and summarise them.



As the figure shown, is the basic review about amazon kindle as shown up, we could see below the different ratings of the product,5-star, 4-star,3-star,2-star,1-star this indicates how the users believe in this certain product. Below that the user has stated the comments about the products also.

Different Types of Sentiment Analysis.

There are four types of sentiment analysis, they are

1. Fine-Grained Sentiment Analysis.

   By segmenting sentiment into additional categories, often highly positive to very negative, fine-grained sentiment analysis delivers a more accurate level of polarity. This can be compared to ratings on a 5-star scale in terms of opinion.

2. Emotion Detection

   Instead of identifying positivity and negative, emotion detection recognises certain emotions. Examples might include astonishment, rage, grief, frustration, and delight.

3. Aspect Based Analysis

   Aspect-based analysis collects the precise component that is being referenced either favourably or unfavourably. For instance, a client may write in a product review that the battery life is too short. The system will then respond that the battery life is the main complaint and not the product.

4. Intent-Based Analysis

   In addition to identifying opinions in a text, intent-based analysis also identifies behaviours. For instance, a frustrated online comment about changing a battery can motivate customer care to get in touch to address that problem.

How Does Sentiment Analysis Works

The several approaches to work on sentiment analysis are,

1. Rule Based Approach

Here, rule-based tokenization, parsing, and the lexicon technique are used.The strategy counts how many positive and negative terms are present in the sample.If there are more positive words than negative words, the emotion is positive; otherwise, it is the opposite.

2. Automatic Approach

This strategy utilises the machine learning method.Predictive analysis is first performed once the datasets have been trained.Word extraction from the text is the subsequent procedure.Different methods, including Naive Bayes, Linear Regression, Support Vector, and Deep Learning, can be used to extract text, just like these machine learning techniques.

3. Hybrid Approach

It is the combination of both the above approaches i.e. rule-based and automatic approach. The surplus is that the accuracy is high compared to the other two approaches.

APPLICATIONS ON SENTIMENT ANALYSIS

Sentiment analysis has numerous uses, including:

1. Social media: Take comments on Instagram as an example. Here, all reviews are examined and divided into three categories: favourable, negative, and neutral.
2. Customer service: Sentiment analysis techniques are used to complete all comments in the Play Store that are rated from 1 to 5.
3. Marketing Sector: In the marketing field, a certain product must be evaluated to determine whether it is good or terrible.
4. Reviewer side: Each reviewer will read the comments, check their accuracy, and provide an overall assessment of the product.

IMPLEMENTATIONS:

Sentiment Analysis on Amazon Kindle review, The first step of implementing the code is by importing the required libraries.

STEP 1: Importing Required Libraries

In [3]:
```
#Data Wrangling
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('white')

#Text Preprocessing
import string
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

#model training and tuning
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split, cross_val_score,GridSearchCV, RandomizedSearchCV
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, roc_auc_score

#ignore the warnings from sklearn
import warnings
warnings.filterwarnings('ignore')
```

STEP 2: Importing Dataset

The dataset which has been used in this project is being taken from Kaggle

https://www.kaggle.com/bharadwaj6/kindle-reviews/home

In [4]:
```
#importing the data
df = pd.read_csv('kindle_reviews.csv')
```

STEP 3: Exploratory Data Analysis

Listed features of datasets is being represented

This dataset provides text reviews on books written by amazon kindle users along with an explicit rating between 1-5. The goal of this notebook is to practice preprocessin text and developing models to predict users sentiment towards the product. Given the explicit scale, the sentiment is divided by ratings where ratings between 3-5 would b classify as Enjoyed (binary one) the book while ratings 1-2 would be classify as disliked (binary zero).

The dataset's features are represented as follows:

asin - ID of the product, like B000FA64PK

helpful - helpfulness rating of the review - example: 2/3.

overall - rating of the product.

reviewText - text of the review (heading).

reviewTime - time of the review (raw).

reviewerID - ID of the reviewer, like A3SPTOKDG7WBLN

reviewerName - name of the reviewer.

summary - summary of the review (description).

unixReviewTime - unix timestamp.

STEP 4:

Listing the information presented in the dataset

In [5]:

```
#Looking at the general information about the dataset. It seems there are some missing reivewText.
df.info()
```

```
RangeIndex: 982619 entries, 0 to 982618
Data columns (total 10 columns):
Unnamed: 0        982619 non-null int64
asin              982619 non-null object
helpful           982619 non-null object
overall           982619 non-null int64
reviewText        982597 non-null object
reviewTime        982619 non-null object
reviewerID        982619 non-null object
reviewerName      978809 non-null object
summary           982618 non-null object
unixReviewTime    982619 non-null int64
dtypes: int64(3), object(7)
memory usage: 75.0+ MB
```

STEP 5:

Checking the null values present in the datasets

In [6]:
```
# 22 missing reviews and needs to be dropped
df.isnull().sum()
```

Out[6]:
```
Unnamed: 0           0
asin                 0
helpful              0
overall              0
reviewText          22
reviewTime           0
reviewerID           0
reviewerName      3810
summary              1
unixReviewTime       0
dtype: int64
```

STEP 6:

Printing the first five values of the data present in the dataset

In [7]:
```
#contained an extra Unnamed column
df.drop(df.columns[0], axis = 1, inplace = True)

#drop the rows where there are no reviews
df.dropna(subset = ['reviewText'], inplace = True)

#changing the reviewTime column to be of datetime type
df.reviewTime = pd.to_datetime(df.reviewTime)

#creating a column with just the year
df['Year'] = df.reviewTime.dt.year
df.head()
```

Out[7]:

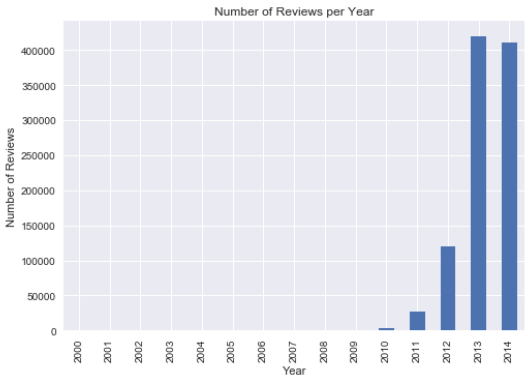
	asin	helpful	overall	reviewText	reviewTime	reviewerID	reviewerName	summary	unixReviewTime	Year
0	B000F83SZQ	[0, 0]	5	I enjoy vintage books and movies so I enjoyed ...	2014-05-05	A1F6404F1VG29J	Avidreader	Nice vintage story	1399248000	2014
1	B000F83SZQ	[2, 2]	4	This book is a reissue of an old one; the auth...	2014-01-06	AN0N05A9LIJEQ	critters	Different...	1388966400	2014
2	B000F83SZQ	[2, 2]	4	This was a fairly interesting read. It had ol...	2014-04-04	A795DMNCJILA6	dot	Oldie	1396569600	2014
3	B000F83SZQ	[1, 1]	5	I'd never read any of the Amy Brewster mysteri...	2014-02-19	A1FV0SX13TWVXQ	Elaine H. Turley "Montana Songbird"	I really liked it.	1392768000	2014
4	B000F83SZQ	[0, 1]	4	If you like period pieces - clothing, lingo, y...	2014-03-19	A3SPTOKDG7WBLN	Father Dowling Fan	Period Mystery	1395187200	2014

STEP 7:

Using Matplotlib for the graphical representations, for number of reviews according year wise

In [9]:

```
#number of reviews per year, it seems almost all reviews came from 2013 and 2014
df.Year.value_counts().sort_index().plot(kind = 'bar')
plt.title('Number of Reviews per Year')
plt.xlabel('Year')
plt.ylabel('Number of Reviews')
plt.show()
```

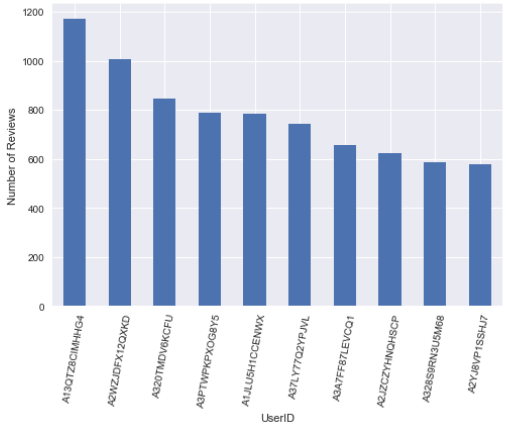


STEP 8:

Representing number of reviews according to the user id

In [10]:

```
#top 10 users base on the number of reviews made
df.reviewerID.value_counts().head(10).plot(kind = 'bar')
plt.xticks(rotation = 80)
plt.xlabel('UserID')
plt.ylabel('Number of Reviews')
plt.show()
```



STEP 9 :

Calculating the mean value for the given data

In [11]:
```
#the overall average review on all the ratings. Seems to be skewed to have above average ratings.
df.overall.mean()
```

Out[11]:
```
4.347791617519695
```

STEP 10:

Printing the overall count values

In [12]:
```
#the disparity between the rating scale is huge, could perform some downsample to have each class roughly have 20k samples
df.overall.value_counts()
```

Out[12]:
```
5    575246
4    254010
3     96193
2     34130
1     23018
Name: overall, dtype: int64
```

STEP 11:

Graphical representation between good rating vs bad ratings.

In [14]:
```
#Take a deeper look into how many ratings we have per scale. Theres are more good ratings than there are bad ratings.
df.overall.value_counts().plot(kind = 'bar')
plt.title('Number of Good Ratings vs Bad Ratings')
plt.xlabel('Rating Scales')
plt.xticks(rotation = 0)
plt.ylabel('Total ratings')
plt.show()
```

STEP 12:

Pre-processing the data.

Text Preprocessing

The following cells provide the steps taken to preprocess the review texts for better feature extraction. Once the text has been preprocessed, it can then be used to develop a vocabulary for training.

1. Removing punctuations
2. Removing non alphabetical words
3. Lowercasing all words
4. Removing stopwords
5. Lemmatization to reduce words to their base form

STEP 13:

Printing the first five values after preprocessing the data

In [19]:
```
#created a dataframe that only contains the reviewText and the overall scoring of item
reviews = df[['reviewText', 'overall']]
reviews.head()
```

Out[19]:

	reviewText	overall
0	I enjoy vintage books and movies so I enjoyed ...	5
1	This book is a reissue of an old one; the auth...	4
2	This was a fairly interesting read. It had ol...	4
3	I'd never read any of the Amy Brewster mysteri...	5
4	If you like period pieces - clothing, lingo, y...	4

STEP 14:

In this step printing the required customer ratings that is good rating and bad ratings listed in the dataset.

In [20]:

```
print('Original Text: ' + str(reviews['reviewText'][1]))
print('\n')

#create an empty mapping table from the str object to strip punctuation from the words
punc = str.maketrans('', '', string.punctuation)
#apply the empty mapping table to each element of the series where x is the review for one document.
reviews['reviewText'] = reviews['reviewText'].apply(lambda x : ' '.join(word.translate(punc) for word in x.split()))
print('Punctuation Remove: ' + str(reviews['reviewText'][1]))
print('\n')

#removing words that is non alpha
reviews['reviewText'] = reviews['reviewText'].apply(lambda x: ' '.join(word for word in x.split() if word.isalpha()))
print('Alphabetical Words: '+ str(reviews['reviewText'][1]))
print('\n')

#making all words to be lowercase
reviews['reviewText'] = reviews['reviewText'].apply(lambda x: ' '.join(word.lower() for word in x.split()))
print('Lowercase Words : '+ str(reviews['reviewText'][1]))
print('\n')

#list of stop words
stop = stopwords.words('english')
#removing the stop words
reviews['reviewText'] = reviews['reviewText'].apply(lambda x : ' '.join(word for word in x.split() if word not in stop))
print('Stopwords Remove: ' + str(reviews['reviewText'][1]))
print('\n')

#Lemmatize words to reduce them to their root form. Note: added the pos = 'v' to reduce the incoming word to verb root
lem = WordNetLemmatizer()
reviews['reviewText'] = reviews['reviewText'].apply(lambda x : ' '.join(lem.lemmatize(word, pos = 'v') for word in x.split()))
print('Lemmatized Text: ' + str(reviews['reviewText'][1]))
```

```
Original Text: This book is a reissue of an old one; the author was born in 1910. It's of the era of, say, Nero Wolfe. The introduction w
as quite interesting, explaining who the author was and why he's been forgotten; I'd never heard of him.The language is a little dated at
times, like calling a gun a "heater."  I also made good use of my Fire's dictionary to look up words like "deshabille" and "Canarsie." St
ill, it was well worth a look-see.


Punctuation Remove: This book is a reissue of an old one the author was born in 1910 Its of the era of say Nero Wolfe The introduction wa
s quite interesting explaining who the author was and why hes been forgotten Id never heard of himThe language is a little dated at times
like calling a gun a 34heater34 I also made good use of my Fires dictionary to look up words like 34deshabille34 and 34Canarsie34 Still i
t was well worth a looksee


Alphabetical Words: This book is a reissue of an old one the author was born in Its of the era of say Nero Wolfe The introduction was qui
te interesting explaining who the author was and why hes been forgotten Id never heard of himThe language is a little dated at times like
calling a gun a I also made good use of my Fires dictionary to look up words like and Still it was well worth a looksee


Lowercase Words : this book is a reissue of an old one the author was born in its of the era of say nero wolfe the introduction was quite
interesting explaining who the author was and why hes been forgotten id never heard of himthe language is a little dated at times like ca
lling a gun a i also made good use of my fires dictionary to look up words like and still it was well worth a looksee


Stopwords Remove: book reissue old one author born era say nero wolfe introduction quite interesting explaining author hes forgotten id n
ever heard himthe language little dated times like calling gun also made good use fires dictionary look words like still well worth looks
ee
```

STEP 15:

Separating The Given ratings into positive and negative, positive as 1 and negative as 2 . and then printing the first five values of the ratings or reviews.

In [22]:

```
#separating the ratings to different sentiment
r1 = reviews[reviews.overall.isin([3,4,5])]
r0 = reviews[reviews.overall.isin([1,2])]
r1.loc[:, 'overall'] = 1
r0.loc[:, 'overall'] = 0

#concat the two new dataframes return one dataframe with preprocessed text and their corresponding labels
rev = pd.concat([r1,r0])
rev.head()
```

Out[22]:

	reviewText	overall
0	enjoy vintage book movies enjoy read book plot...	1
1	book reissue old one author bear era say nero ...	1
2	fairly interest read old style terminologyi gl...	1
3	id never read amy brewster mysteries one reall...	1
4	like period piece clothe lingo enjoy mystery a...	1

CHALLENGES FACED ON SENTIMENT ANALYSIS:

Major obstacles face sentiment analysis approach:

1. It is very challenging to determine whether a comment is optimistic or pessimistic when the data is presented in the form of a tone.
2. You must determine if the data is beneficial or negative if it is shown as an emoji.
3. Even detecting ironic, caustic, or comparative comments is difficult.
4. A neutral statement is difficult to compare.

CONCLUSION:

Sentiment Analysis on Amazon Kindle Review, States that the getting different reviews about the product such as positive or negative, also the different ratings, like 5- star, 4-star, 3-star,2-star,1-star also every analysis has been done and some of the challenges is sometimes it is difficult to determine the data in form of tone, determining the data in the form of emoji , and detecting the comparative comment is difficult, when an neutral statement is difficult to compare, the sentiment analysis is an NLP technique used for the organization for finding the general review of their products.

REFERENCES:

1. Levent Guner, Emilie Coyne and Jim Smit, “Sentiment Analysis of Amazon.com Reviews”, March,2019,Available:https://www.researchgate.net/publication/332622380_Sentiment_analysis_for_Amazoncom_reviews.

2. Xing Fang and Justin Zhan, “Sentiment Analysis using product review data”, Journal of Big Data, vol. 2, no.1, 16 June 2015. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2

3. Wanliang Tan, Xinyu Wang, and Xinyu Xu,“Sentiment Analysis for Amazon Reviews”, Available:http://cs229.stanford.edu/proj2018/report/122.pdf

4. Callen Rain, “Analysis in Amazon Reviews Using Probabilistic Machine Learning”, 2012.Available: https://www.semanticscholar.org/paper/Analysis-in-Amazon-Reviews-Using-Probabilistic-Rain/f0afe9ea9d286248336ee9dc4e954aecde3475b

5. Nishit Shrestha, Fatma Nasoz, “Deep Learning Sentiment Analysis of Amazon.com Reviews and Ratings”,International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.8, No.1, February 2019. Available: https://arxiv.org/abs/1904.04096

6. sklearn.metrics.precision_recall_fscore_supportAvailable:https://scikit-earn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

7. https://www.geeksforgeeks.org/what-is-sentiment-analysis/