NLP through K.G.F. Chapter-2 Movie IMDb User Review

In this notebook the User Review of the movie KGF Chapter-2 in the IMDb website is used to train the classification model for classifying whether the review is negative ,neutral or positive.The data that is to be trained undergoes various stages of NLP , removing html from the text , removing punctuation,tokenizing,stemming etc..Here Naive Bayes Classifier ,Decision Tree Classifier and Random Forest Classifier are used for training the model.

I have used Octoparse tool for webscarping the IMDb data.

In [1]:
```
# Connecting google drive to this notebook
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

Loading the dataset and finding some insights from the data.

In [2]:
```
# importing libraries
import pandas as pd
import numpy as np
import nltk
from bs4 import BeautifulSoup
import string
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from nltk.stem.porter import PorterStemmer
```

In [3]:
```
# importing the dataset
data = pd.read_csv('/content/drive/MyDrive/KGF2 data.csv')
data.head()
```

Out[3]:

	Title	Score	Name	Date	actions	text
0	The Finest Part 2 ever\n	10.0	sammyaklagade	14 April 2022	\n 182 out of 211 found thi...	This will go down as the greatest Indian films...
1	Paisa Vasool movie - KGF Chapter 2\n	10.0	arjuncshekartheofficial	14 April 2022	\n 141 out of 168 found thi...	KGF Chapter 1 movie was just a Trailer, But th...
2	KGF 2 Box Office Tsunami Is Coming\n	10.0	rosheenkan	14 April 2022	\n 87 out of 107 found this...	As expected, KGF 2 is getting positive respons...
3	Goosebumps overloaded!!\n	9.0	sriramthestranger	14 April 2022	\n 92 out of 115 found this...	The screenplay, cuts and sound effects deserve...
4	Expectations surpassed!\n	10.0	Prashast_Singh	15 April 2022	\n 69 out of 86 found this ...	"Hindi viewers, please don't regret later on w...

In [4]:
```
# checking columns of the dataset
data.columns
```

Out[4]:
```
Index(['Title', 'Score', 'Name', 'Date', 'actions', 'text'], dtype='object')
```

In [5]:
```
# checking shape of the dataset
data.shape
```

Out[5]:
```
(1079, 6)
```

In [6]:
```
# checking missing values of the dataset
data.isnull().sum()
```

Out[6]:
```
Title      0
Score      5
Name       0
Date       0
actions    0
text       0
dtype: int64
```

In [7]:
```
# creating summary of the dataset
data.describe(include='all')
```

Out[7]:

	Title	Score	Name	Date	actions	text
count	1079	1074.000000	1079	1079	1079	1079
unique	1015	NaN	1079	5	121	1062
top	Awesome\n	NaN	sammyaklagade	14 April 2022	\n 0 out of 0 found this he...	This will go down as the greatest Indian films...
freq	8	NaN	1	440	566	6
mean	NaN	9.304469	NaN	NaN	NaN	NaN
std	NaN	1.998471	NaN	NaN	NaN	NaN
min	NaN	1.000000	NaN	NaN	NaN	NaN
25%	NaN	10.000000	NaN	NaN	NaN	NaN
50%	NaN	10.000000	NaN	NaN	NaN	NaN
75%	NaN	10.000000	NaN	NaN	NaN	NaN
max	NaN	10.000000	NaN	NaN	NaN	NaN

In [8]:
```
# creating overview of the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1079 entries, 0 to 1078
Data columns (total 6 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   Title    1079 non-null   object 
 1   Score    1074 non-null   float64
 2   Name     1079 non-null   object 
 3   Date     1079 non-null   object 
 4   actions  1079 non-null   object 
 5   text     1079 non-null   object 
dtypes: float64(1), object(5)
memory usage: 50.7+ KB
```

In [9]:
```
# evaluating frequency of score
data.Score.value_counts()
```

Out[9]:
```
10.0    889
9.0      62
1.0      36
8.0      23
6.0      18
7.0      16
4.0      11
2.0       8
5.0       6
3.0       5
Name: Score, dtype: int64
```

In [10]:
```
# creating a table with columns as Score and grouped by date
pd.crosstab(data['Date'],data['Score'])
```

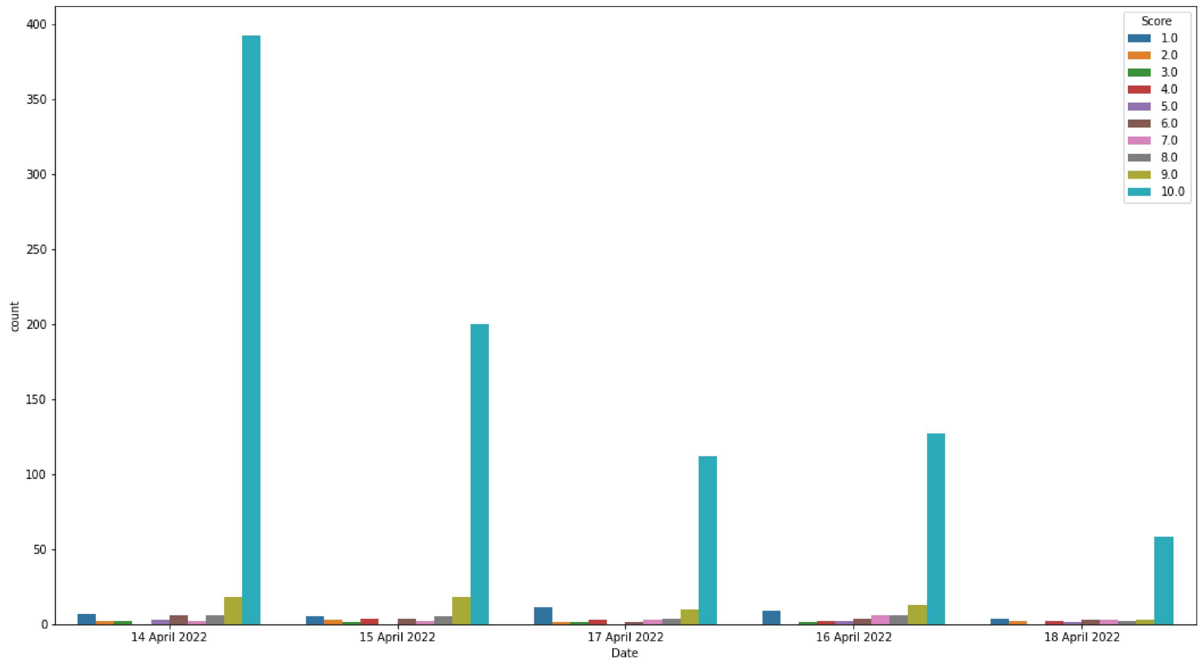
Out[10]:

Score	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
Date										
14 April 2022	7	2	2	0	3	6	2	6	18	392
15 April 2022	5	3	1	4	0	4	2	5	18	200
16 April 2022	9	0	1	2	2	4	6	6	13	127
17 April 2022	11	1	1	3	0	1	3	4	10	112
18 April 2022	4	2	0	2	1	3	3	2	3	58

In [11]:
```
# Plotting a graph
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(18,10))
sns.countplot(x="Date",hue="Score",data=data)
```

Out[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff6ae4b7090>
```



Preparing the dataset for training the various classification model.

In [12]:
```
# dropping useless columns from the dataframe
data_1 = data.drop(['Title','Name','Date','actions'],axis=1)
```

In [13]:
```
# dropping rows from the dataframe with null values
data_1 = data_1.dropna(axis=0)
```

In [14]:
```
# checking shape of the dataframe
data_1.shape
```

Out[14]:
```
(1074, 2)
```

In [15]:
```
# looking first 10 rows of the dataframe
data_1.head(10)
```

Out[15]:

	Score	text
0	10.0	This will go down as the greatest Indian films...
1	10.0	KGF Chapter 1 movie was just a Trailer, But th...
2	10.0	As expected, KGF 2 is getting positive respons...
3	9.0	The screenplay, cuts and sound effects deserve...
4	10.0	"Hindi viewers, please don't regret later on w...
5	10.0	Watched 7 am show.\n\nCan't compare with any H...
6	10.0	KGF - Chapter 1, set the tone for a slick gang...
7	9.0	KFF part 2- Boisterous, loud and machismo type...
8	10.0	It is one of the best movies and a history for...
9	10.0	Blockbuster movie.. Wellmade action sequences....

Classifying Score into 3 categories negative ,neutral, positive

- 1 to 4 --> 0(**negative**)
- 5 to 7 --> 1(**neutral**)
- 8 to 10 --> 2(**positive**)

In [ ]:
```
# classifying score into 3 categories

x=0
for i in data_1['Score']:
  if i>7:
    i=2
  elif i<8 and i>4:
    i=1
  else:
    i=0
  data_1['Score'][x]=i
  x=x+1
```

In [17]:
```
# removing some end rows from the dataframe
data_1=data_1.drop(labels=range(1074,1079),axis=0)
```

In [18]:
```
# looking last 10 rows of the dataframe
data_1.tail(10)
```

Out[18]:

	Score	text
1064	2.0	EXCELLENT ACTION PACKED MOVIE.\n\nElevation is...
1065	2.0	This movie has even surpassed the hype we had ...
1066	2.0	KGF chapter2 is not just a movie it's an emoti...
1067	2.0	Emotions + Elevations + Action!!\n\nThat's KGF...
1068	2.0	Brilliant screenplay and bgm.\n\nAs a villain ...
1069	2.0	Worth waiting for 4 years, worth every penny. ...
1070	1.0	Good choreography, direction, cinematography, ...
1071	2.0	The screenplay, cuts and sound effects deserve...
1072	2.0	Everything was spot on, good movie - just a li...
1073	2.0	No doubt its a masterpiece Next level movie Ro...

In [19]:
```
# removing html text from the dataframe
def remove_htmltext(text):
  soup = BeautifulSoup(text,'lxml')
  html_free=soup.get_text()
  return html_free

data_1['text']=data_1['text'].apply(lambda x: remove_htmltext(x))

data_1.head()
```

Out[19]:

	Score	text
0	2.0	This will go down as the greatest Indian films...
1	2.0	KGF Chapter 1 movie was just a Trailer, But th...
2	2.0	As expected, KGF 2 is getting positive respons...
3	2.0	The screenplay, cuts and sound effects deserve...
4	2.0	"Hindi viewers, please don't regret later on w...

In [20]:
```
# removing punctuation from the dataframe
def remove_punctuation(text):
  no_punct = "".join([c for c in text if c not in string.punctuation])
  return no_punct

data_1['text']=data_1['text'].apply(lambda x:remove_punctuation(x))

data_1.head()
```

Out[20]:

	Score	text
0	2.0	This will go down as the greatest Indian films...
1	2.0	KGF Chapter 1 movie was just a Trailer But thi...
2	2.0	As expected KGF 2 is getting positive response...
3	2.0	The screenplay cuts and sound effects deserves...
4	2.0	Hindi viewers please dont regret later on when...

In [21]:
```
# tokenizing the string of words into single word and changing into lower case
tokenizer = RegexpTokenizer(r'\w+')

data_1['text']=data_1['text'].apply(lambda x: tokenizer.tokenize(x.lower()))

data_1.head()
```

Out[21]:

	Score	text
0	2.0	[this, will, go, down, as, the, greatest, indi...
1	2.0	[kgf, chapter, 1, movie, was, just, a, trailer...
2	2.0	[as, expected, kgf, 2, is, getting, positive, ...
3	2.0	[the, screenplay, cuts, and, sound, effects, d...
4	2.0	[hindi, viewers, please, dont, regret, later, ...

In [22]:
```
# removing the stopwords from the dataframe
nltk.download('stopwords')
def remove_stopwords(text):
  words = [w for w in text if w not in set(stopwords.words('english'))]
  return words

data_1['text']=data_1['text'].apply(lambda x:remove_stopwords(x))

data_1.head()
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[22]:

	Score	text
0	2.0	[go, greatest, indian, films, ever, storyline,...
1	2.0	[kgf, chapter, 1, movie, trailer, movie, feast...
2	2.0	[expected, kgf, 2, getting, positive, response...
3	2.0	[screenplay, cuts, sound, effects, deserves, s...
4	2.0	[hindi, viewers, please, dont, regret, later, ...

In [23]:
```
# applying stemming to the dataframe
stemmer = PorterStemmer()
def word_stemmer(text):
  stem_text = " ".join([stemmer.stem(i) for i in text])
  return stem_text

data_1['text']=data_1['text'].apply(lambda x:word_stemmer(x))

data_1.head()
```

Out[23]:

	Score	text
0	2.0	go greatest indian film ever storylin sublim e...
1	2.0	kgf chapter 1 movi trailer movi feast prashant...
2	2.0	expect kgf 2 get posit respons sure there poss...
3	2.0	screenplay cut sound effect deserv stand ovat ...
4	2.0	hindi viewer pleas dont regret later itll play...

In [24]:
```
# creating the Bag of Words model
corpus = []
corpus = data_1['text']
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = data_1.iloc[:,0].values
```

In [25]:
```
# printing the predictor variable
print(X)
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

In [26]:
```
# printing the target variable
print(y)
```

```
[2. 2. 2. ... 2. 2. 2.]
```

Splitting the dataset into training and test set.

In [27]:
```
# splitting the dataset into training set and test set
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_stat
e=0)
```

Training the dataset using classification model and analyzing accuracy of the model.

In [28]:
```
# training the Naive Bayes Classification model to the training dataset
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train,y_train)
```

Out[28]:
```
GaussianNB()
```

In [29]:
```
# predicting the values classified by the following model
y_pred = classifier.predict(X_test)
```

In [30]:
```
# calculating accuracy of the model by accuracy_score
from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test,y_pred)
acc
```

Out[30]:
```
0.7990654205607477
```

In [31]:
```
# calculating accuracy of the model by confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[  1   0   8]
 [  0   0   6]
 [ 28   1 170]]
```

In [32]:
```
# training the Decision tree Classification model
from sklearn.tree import DecisionTreeClassifier
classifier_1 = DecisionTreeClassifier(criterion="entropy",random_state=0)
classifier_1.fit(X_train,y_train)
```

Out[32]:
```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

In [33]:
```
# predicting the values classified by the following model
y_pred_1 = classifier_1.predict(X_test)
```

In [34]:
```
# calculating accuracy of the model by accuracy_score
acc_1 = accuracy_score(y_test,y_pred_1)
print(acc_1)
```

```
0.8644859813084113
```

In [35]:
```
# calculating accuracy of the model by confusion matrix
cm_1 = confusion_matrix(y_test,y_pred_1)
cm_1
```

Out[35]:
```
array([[  0,   1,   8],
       [  0,   1,   5],
       [ 10,   5, 184]])
```

In [36]:
```
# training the random forest classification model
from sklearn.ensemble import RandomForestClassifier
classifier_2 = RandomForestClassifier(n_estimators=10,criterion="entropy",rand
om_state=0)
classifier_2.fit(X_train,y_train)
```

Out[36]:
```
RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)
```

In [37]:
```
# predicting the values classified by the following model
y_pred_2 = classifier_2.predict(X_test)
```

In [38]:
```
# calculating accuracy of the model by accuracy_score
acc_2 = accuracy_score(y_test,y_pred_2)
acc_2
```

Out[38]:
```
0.9252336448598131
```

In [39]:
```
# calculating accuracy of the model by confusion matrix
cm_2 = confusion_matrix(y_test,y_pred_2)
cm_2
```

Out[39]:
```
array([[  0,   0,   9],
       [  0,   0,   6],
       [  1,   0, 198]])
```

Analyzing various classification models through accuracy_score and confusion matrix it is found that Randomforestclassifier model yields high accuracy.