Objective : In the present scenario, a person wants ease in their lives, so E-commerce has become a great and admirable involvement in providing the availability of any product at the doorsteps. But how a person can know the efficiency and originality of the product just by looking at the pictures and the details of the product on the websites. To overcome these issues the E-commerce websites have introduced the concept of the Reviews. Reviews are written by the customers who have already purchased it. Studies show that Product reviews are one of the most important points one considers during the purchasing from E-commerce websites like Flipkart, Snapdeal, Amazon and so on. This paper proposes a model that detects whether the given review is positive, negative, or neutral using the method of sentiment analysis. And using Data Analysis we can find the extension of this paper, we are planning to use a type of sentiment analysis, Opinion Mining which is the research field that predominantly makes automatic systems that will find opinion from the text written in human language. Using opinion mining, we can find whether the given reviews are fake or not. In this paper we have used Amazon food reviews data and based on the rating given by the user we are classifying reviews as positive, negative, or neutral. For positive review ratings given were 4 and 5. For negative review ratings given were 1 and 2. For neutral, rating given was 3.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import csv
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,mean_squared_error,confusion_matrix
%matplotlib inline

csv_file = '/content/Reviews.csv'

Data = pd.read_csv(csv_file, error_bad_lines=False, engine="python")
```

```
/usr/local/lib/python3.8/dist-packages/IPython/core/interactiveshell.py:3326: FutureWarning: The error_bad_lines argument has been

  exec(code_obj, self.user_global_ns, self.user_ns)
Skipping line 40252: unexpected end of data
```

```
Data.head()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	Helpful
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	

The Score column of this dataset contains the ratings that customers have given to the product based on their experience with the product.

EDA : Drop duplicate and drop null

```
dup = Data[Data.duplicated(subset=['UserId','ProfileName','Time','Text'],keep=False)]
dup.shape
```

```
(5023, 10)
```

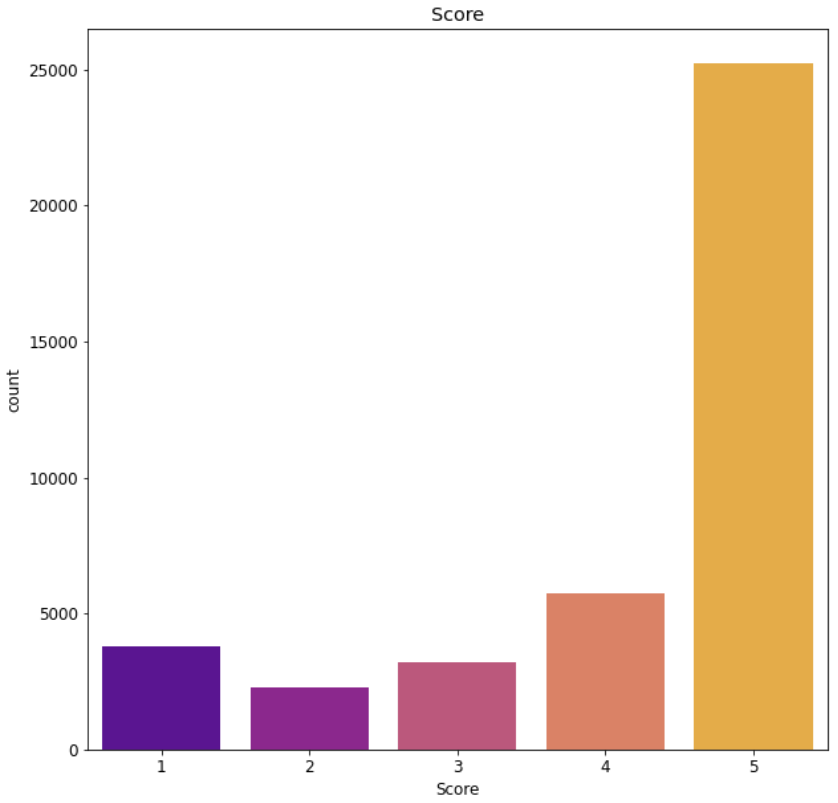
```
data = Data.drop_duplicates(subset=['UserId','ProfileName','Time','Text'],keep='first',inplace=False)
data.shape
```

```
(37700, 10)
```

Visualize the score

```
sns.countplot(data['Score'], palette="plasma")
fig = plt.gcf()
fig.set_size_inches(10,10)
plt.title('Score')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: P
  warnings.warn(
Text(0.5, 1.0, 'Score')
```

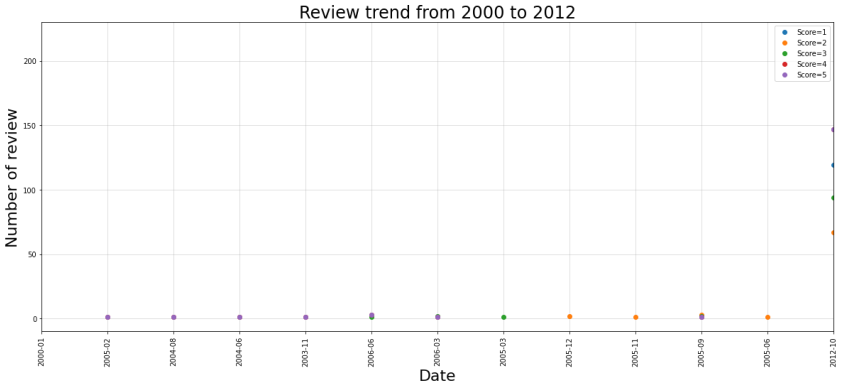


Analyisis of Reviews over time

```
data['date'] = pd.to_datetime(data['Time'],unit='s')
dataf = data[['date','Text','Score']]
dataf.date = data.date.dt.strftime('%Y-%m')
# dataf['date'] = dataf['date'].dt.to_timestamp()
dataf = dataf.sort_values(by=['date']).reset_index(drop=True)
dataf_1 = dataf[dataf['Score'] == 1]
dataf_2 = dataf[dataf['Score'] == 2]
dataf_3 = dataf[dataf['Score'] == 3]
dataf_4 = dataf[dataf['Score'] == 4]
dataf_5 = dataf[dataf['Score'] == 5]

dataf_1 = dataf_1.groupby('date')['Score'].count().reset_index()
dataf_2 = dataf_2.groupby('date')['Score'].count().reset_index()
dataf_3 = dataf_3.groupby('date')['Score'].count().reset_index()
dataf_5 = dataf_4.groupby('date')['Score'].count().reset_index()
dataf_4 = dataf_4.groupby('date')['Score'].count().reset_index()

plt.figure(figsize=(20,8))

plt.plot_date(x=dataf_1['date'],y=dataf_1['Score'],label='Score=1')
plt.plot_date(x=dataf_2['date'],y=dataf_2['Score'],label='Score=2')
plt.plot_date(x=dataf_3['date'],y=dataf_3['Score'],label='Score=3')
plt.plot_date(x=dataf_4['date'],y=dataf_4['Score'],label='Score=4')
plt.plot_date(x=dataf_5['date'],y=dataf_5['Score'],label='Score=5')
plt.grid(linewidth=0.5,alpha=0.75)
plt.xticks(rotation=90)
plt.xlim('2000-01','2012-10')
plt.xlabel('Date',fontsize=22)
plt.ylabel('Number of review',fontsize=22)
plt.title('Review trend from 2000 to 2012',fontsize=24);
plt.savefig('review_trend.png')
```

```
plt.legend()
plt.show()
```

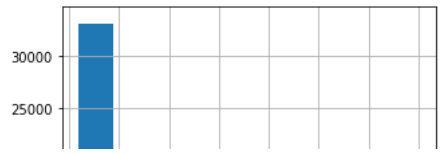
```
<ipython-input-40-952c28c7cf63>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  data['date'] = pd.to_datetime(data['Time'],unit='s')
/usr/local/lib/python3.8/dist-packages/pandas/core/generic.py:5516: SettingWithCop
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  self[name] = value
```



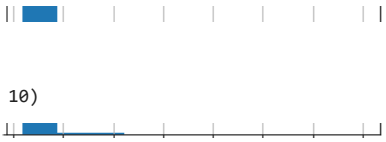

```
purchases = data[['ProductId','UserId']].groupby('UserId').agg({'ProductId': ['count']})
purchases.columns = ['No_of_products_purchased']
purchases = purchases.reset_index()
purchases.head(2)
```

	UserId	No_of_products_purchased
0	#oc-R119LM8D59ZW8Y	1
1	#oc-R13X3YIJ6GLT0C	1

```
plt.figure(figsize=(5,5))
purchases['No_of_products_purchased'].hist()
plt.xlabel('No of purchases')
plt.ylabel('No of users')
plt.show()
print(purchases['No_of_products_purchased'].describe())
```



Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.



```
Data.shape
```

```
(40250, 10)
```

```
Data.describe()
```

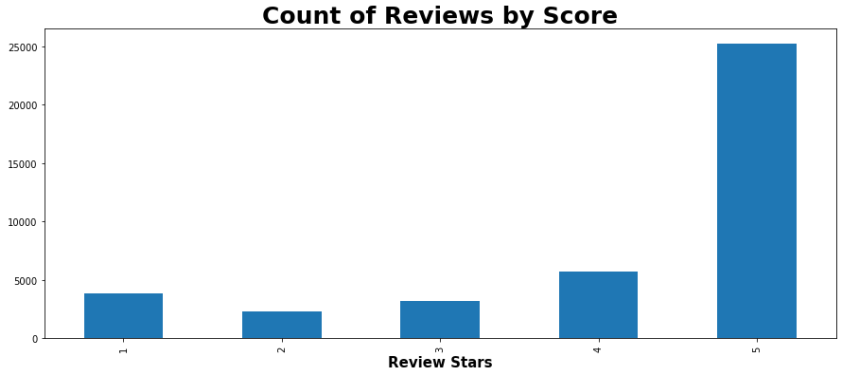
	Id	HelpfulnessNumerator	HelpfulnessDenominator	Score	
count	40250.000000	40250.000000	40250.000000	40250.000000	4.
mean	20125.500000	1.614559	2.069988	4.149988	1.
std	11619.318504	5.760170	6.348798	1.327692	4.
min	1.000000	0.000000	0.000000	1.000000	9.
25%	10063.250000	0.000000	0.000000	4.000000	1.
50%	20125.500000	0.000000	1.000000	5.000000	1.
75%	30187.750000	2.000000	2.000000	5.000000	1.
max	40250.000000	398.000000	401.000000	5.000000	1.

```
data = Data.dropna()
```

```
data.shape
```

```
(40247, 10)
```

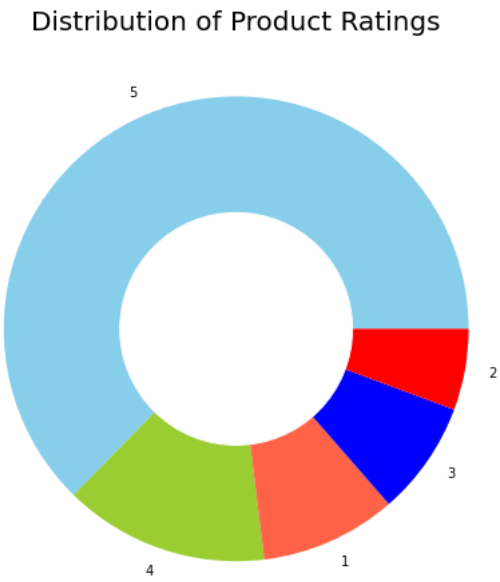
```
ax = data['Score'].value_counts().sort_index() .plot(kind='bar',figsize=(15, 6))
ax.set_xlabel('Review Stars',fontsize=15,fontweight='bold')
ax.set_title('Count of Reviews by Score',fontsize=25,fontweight='bold')
plt.show()
```



Sentiment Analysis : The Score column of this dataset contains the ratings that customers have given to the product based on their experience with the product. So let's take a look at the rating breakdown to see how most customers rate the products they buy

```
ratings = data["Score"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue", "red"]
plt.figure(figsize=(10, 8))
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
```

```
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Distribution of Product Ratings", fontsize=20)
plt.show()
```




```
#sns.pairplot(data)
```

According to the figure above, more than half of people rated products they bought with 5 stars, which is good. Now, I'm going to add three more columns to this dataset as Positive, Negative, and Neutral by calculating the sentiment scores of the customer reviews mentioned in the Text column of the dataset:

How to determine if a review is positive or negative?

We could use Score/Rating. A rating of 4 or 5 can be considered as a positive review. A rating of 1 or 2 can be considered as a negative one. A review of rating 3 is considered neutral

VADER Sentiment analysis usually adopted on social media platforms to express one's sentiment, which makes it a great fit for social media sentiment text analysis. VADER has the advantage of assessing the sentiment of any given text without the need for previous training as we might have to for Machine Learning models.

```
import nltk
nltk.downloader.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
True
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
```

```
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i in data["Text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i in data["Text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i in data["Text"]]
print(data.head())
```

```
<ipython-input-52-65648d19a827>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i in data["Text"]]
<ipython-input-52-65648d19a827>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i in data["Text"]]
   Id   ProductId          UserId                      ProfileName  \
0   1  B001E4KFG0  A3SGXH7AUHU8GW                       delmartian
1   2  B00813GRG4  A1D87F6ZCVE5NK                           dll pa
2   3  B000LQOCH0   ABXLMWJIXXAIN  Natalia Corres "Natalia Corres"
```

```
3   4  B000UA0QIQ  A395BORC6FGVXV                           Karl
4   5  B006K2ZZ7K  A1UQRSCLF8GW1T    Michael D. Bigham "M. Wassir"

   HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
0                     1                       1      5  1303862400
1                     0                       0      1  1346976000
2                     1                       1      4  1219017600
3                     3                       3      2  1307923200
4                     0                       0      5  1350777600

                Summary                                       Text  \
0  Good Quality Dog Food  I have bought several of the Vitality canned d...
1      Not as Advertised  Product arrived labeled as Jumbo Salted Peanut...
2  "Delight" says it all  This is a confection that has been around a fe...
3         Cough Medicine  If you are looking for the secret ingredient i...
4            Great taffy  Great taffy at a great price.  There was a wid...

   Positive  Negative  Neutral
0     0.305     0.000    0.695
1     0.000     0.138    0.862
2     0.155     0.091    0.754
3     0.000     0.000    1.000
4     0.448     0.000    0.552
<ipython-input-52-65648d19a827>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i in data["Text"]]
```

So we can say that most of the reviews of the products available on Amazon are positive, as the total sentiment scores of Positive and Neural are much higher than Negative scores.

Summary

So this is how we can analyze the sentiments of the product reviews at amazon. There are so many customers buying products from Amazon that today Amazon earns an average of $ 638.1 million per day. So having such a large customer base, it will turn out to be an amazing data science project if we can analyze the sentiments of Amazon product reviews. I hope you liked this article on Amazon Product Reviews Sentiment Analysis with Python. Feel free to ask your valuable questions in the comments section below.