

## Experiment 3 - Cognitive Computing Application using Python

Roll No.	37
Name	Mikil Lalwani
Class	D20 B
Subject	Data Science Lab
LO Mapped	LO2: Explore use cases of Cognitive Computing.

**Aim:** To Build a cognitive computing application on text data using Python.

### **Cognitive Computing:**

Cognitive computing is a subfield of artificial intelligence (AI) that refers to systems that aim to simulate human thought processes and reasoning in order to interpret data, understand natural language, and learn from interactions. The goal is to create intelligent systems that can adapt and make decisions similar to the way humans would.

### **Examples of Real-World Use Cases of Cognitive Computing**

IBM Watson has been utilized in the healthcare industry to analyze medical records and find insights to improve diagnosis and treatment plans. It can understand patient history, current conditions and latest research to provide recommended actions.

Virtual assistants like Alexa, Siri and Google Assistant use cognitive computing to understand natural speech, search vast knowledge bases and respond appropriately in conversational manner. Financial institutions use cognitive computing systems to monitor economic conditions, analyze financial transactions and discover patterns of fraud and risk. This improves compliance and protects against threats.

Retailers like Amazon and Netflix apply cognitive computing to understand purchasing history and interests of customers. Product recommendations and targeted promotions can then be presented.

### **What are the Benefits of Cognitive Computing?**

Cognitive computing has several benefits:

- Improved decision making. By analyzing vast amounts of data, cognitive computing can provide insights that help in making informed decisions.
- Increased efficiency. Cognitive computing systems can automate complex tasks, leading to increased efficiency and productivity.
- Personalized experience. Cognitive computing can provide a personalized experience to users by understanding their preferences and behaviors.

### **What are the Challenges of Cognitive Computing?**

Despite its benefits, cognitive computing faces several challenges:

- Data privacy. Cognitive computing systems require large amounts of data, which raises concerns about data privacy and security.
- Complexity. The development and implementation of cognitive computing systems are complex and require significant resources.
- Lack of understanding. There is a lack of understanding and acceptance of cognitive computing among many people, which can hinder its adoption.

### **Ethical Considerations of Cognitive Computing**

Cognitive computing, like other AI technologies, raises several ethical considerations. These include concerns about job displacement due to automation, the transparency of AI decision-making processes, and the potential misuse of AI technologies. It's crucial to address these issues through regulations and ethical guidelines to ensure the responsible use of cognitive computing.

## **Sentiment Analysis:**

### **What is Sentiment Analysis?**

Sentiment analysis is the process of analyzing digital text to determine if the emotional tone of the message is positive, negative, or neutral. Today, companies have large volumes of text data like emails, customer support chat transcripts, social media comments, and reviews. Sentiment analysis tools can scan this text to automatically determine the author's attitude towards a topic. Companies use the insights from sentiment analysis to improve customer service and increase brand reputation.

### **Why is sentiment analysis important?**

Sentiment analysis, also known as opinion mining, is an important business intelligence tool that helps companies improve their products and services. We give some benefits of sentiment analysis below.

#### **1. Provide objective insights**

Businesses can avoid personal bias associated with human reviewers by using artificial intelligence (AI)-based sentiment analysis tools. As a result, companies get consistent and objective results when analyzing customers' opinions.

For example, consider the following sentence:

*I'm amazed by the speed of the processor but disappointed that it heats up quickly.*

Marketers might dismiss the discouraging part of the review and be positively biased towards the processor's performance. However, accurate sentiment analysis tools sort and classify text to pick up emotions objectively.

#### **2. Build better products and services**

A sentiment analysis system helps companies improve their products and services based on genuine and specific customer feedback. AI technologies identify real-world objects or situations (called entities) that customers associate with negative sentiment. From the above example, product engineers focus on improving the processor's heat management capability because the text analysis software associates disappointment (negative) with processor (entity) and heats up (entity).

#### **3. Analyze at scale**

Businesses constantly mine information from a vast amount of unstructured data, such as emails, chatbot transcripts, surveys, customer relationship management records, and product feedback. Cloud-based sentiment analysis tools allow businesses to scale the process of uncovering customer emotions in textual data at an affordable cost.

#### 4. Real-time results

Businesses must be quick to respond to potential crises or market trends in today's fast-changing landscape. Marketers rely on sentiment analysis software to learn what customers feel about the company's brand, products, and services in real time and take immediate actions based on their findings. They can configure the software to send alerts when negative sentiments are detected for specific keywords.

### **Python Library Function Used:**

#### 1. Pandas-

Pandas is a popular Python library for data manipulation and analysis. It provides powerful data structures, like DataFrames and Series, that make it easy to work with structured data. Pandas simplifies tasks such as loading data from various sources, cleaning and transforming data, and performing operations like filtering, grouping, and aggregating. It also supports handling missing data and time series data efficiently. With its versatile functionality, Pandas is widely used in data science, enabling tasks like data exploration, visualization, and preparation for machine learning. Its concise syntax and extensive documentation make it a valuable tool for anyone working with tabular data in Python.

#### 2. Matplotlib-

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations in various formats. It provides a wide range of functions and tools for creating plots, charts, and graphs, making it a go-to choice for data visualization and scientific plotting. Matplotlib's customizable and publication-quality graphics support various plot types, including line plots, scatter plots, bar charts, histograms, and more. It offers flexibility to control every aspect of the visualization, from axis labels to colors and styles. Matplotlib can be used both as a standalone library and in conjunction with other libraries, such as NumPy and Pandas, to visualize data effectively.

#### 3. Seaborn:

Seaborn is a data visualization library based on Matplotlib. It specializes in creating attractive and informative statistical graphics. Seaborn simplifies the process of generating complex visualizations with minimal code. It provides a high-level interface for creating various types of statistical plots, such as bar plots, scatter plots, heatmaps, and more. Seaborn also offers built-in themes and color palettes to enhance the aesthetics

of your visualizations. It's commonly used for exploring data and gaining insights through graphical representation.

4. NLTK (Natural Language Toolkit):

NLTK is a comprehensive library for natural language processing (NLP) in Python. It offers tools and resources for working with text data, including tasks like tokenization, stemming, lemmatization, part-of-speech tagging, and sentiment analysis. NLTK also includes corpora and lexical resources, making it a valuable resource for linguistic research and text analysis. It's widely used in NLP applications, such as text classification, language modeling, and information retrieval.

### **Code and Observation:**

**Dataset -** <https://www.kaggle.com/datasets/vivekprajapati2048/amazon-customer-reviews>

1. Import the required libraries.

```
[20]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
```

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

```
[20]: True
```

2. Read the dataset.

```
[21]: df = pd.read_csv('Reviews.csv')
      df.head()
```

```
[21]:
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

### 3. Check the dataset.

```
[22]: df.shape
```

```
[22]: (568454, 10)
```

```
[23]: df.info()
```

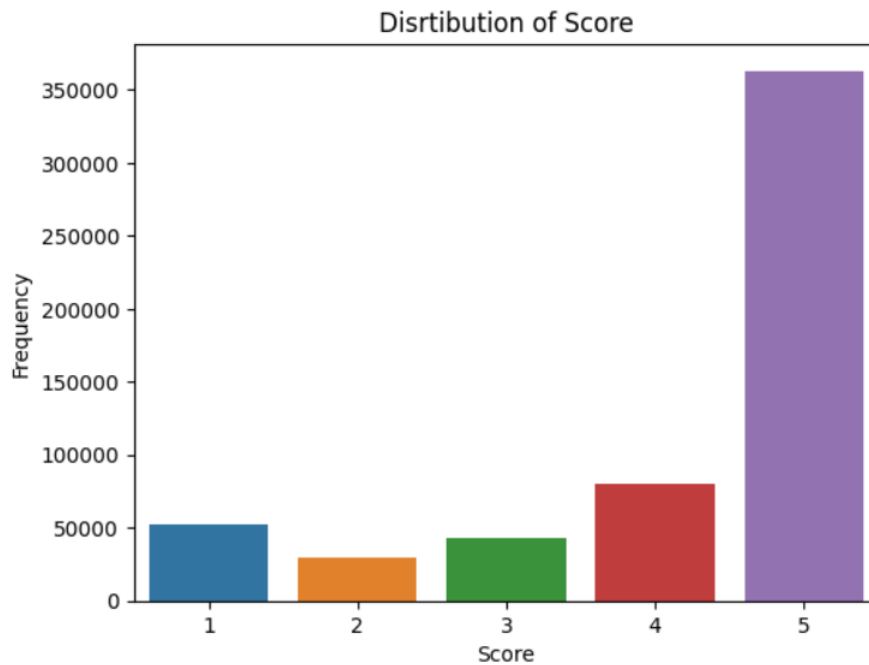
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Id                                    568454 non-null  int64
1   ProductId                           568454 non-null  object
2   UserId                               568454 non-null  object
3   ProfileName                          568428 non-null  object
4   HelpfulnessNumerator                 568454 non-null  int64
5   HelpfulnessDenominator               568454 non-null  int64
6   Score                                568454 non-null  int64
7   Time                                 568454 non-null  int64
8   Summary                              568427 non-null  object
9   Text                                 568454 non-null  object
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
```

```
[24]: df['Score'].value_counts()
```

```
[24]: Score
5    363122
4    80655
1    52268
3    42640
2    29769
Name: count, dtype: int64
```

## 4. Graph of the distribution of the reviews.

```
[25]: sns.countplot(data=df, x='Score')  
      plt.ylabel('Frequency')  
      plt.title('Disrtibution of Score')  
      plt.show()
```



## 5. Import and use the sentiment analyser.

```
[26]: from nltk.sentiment import SentimentIntensityAnalyzer  
  
      sia = SentimentIntensityAnalyzer()
```

```
[27]: sia.polarity_scores('Best product')
```

```
[27]: {'neg': 0.0, 'neu': 0.192, 'pos': 0.808, 'compound': 0.6369}
```

```
[28]: sia.polarity_scores('Quality is not good')
```

```
[28]: {'neg': 0.445, 'neu': 0.555, 'pos': 0.0, 'compound': -0.3412}
```

```
[31]: res = {}
      for i in range(df.shape[0]):
          text = df['Text'].iloc[i]
          myid = df['Id'].iloc[i]

          res[myid] = sia.polarity_scores(text)
```

## 6. Use Vaders to

```
[32]: vaders = pd.DataFrame(res).T
      vaders = vaders.reset_index().rename(columns={'index': 'Id'})
      vaders = vaders.merge(df, how='left')
```

```
[33]: vaders.head()
```

	Id	neg	neu	pos	compound	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time
--	----	-----	-----	-----	----------	-----------	--------	-------------	----------------------	------------------------	-------	------

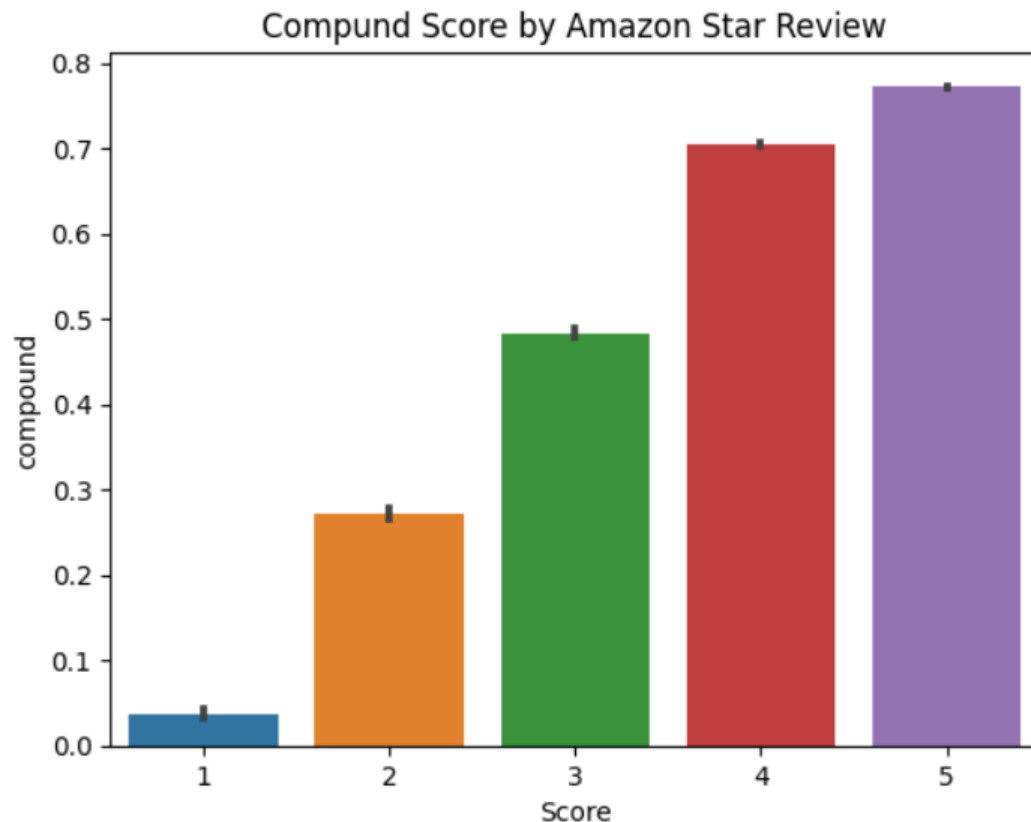
0	1	0.000	0.695	0.305	0.9441	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400
---	---	-------	-------	-------	--------	------------	----------------	------------	---	---	---	------------

1	2	0.138	0.862	0.000	-0.5664	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000
---	---	-------	-------	-------	---------	------------	----------------	--------	---	---	---	------------

## 7. Plot of the sentiment score.

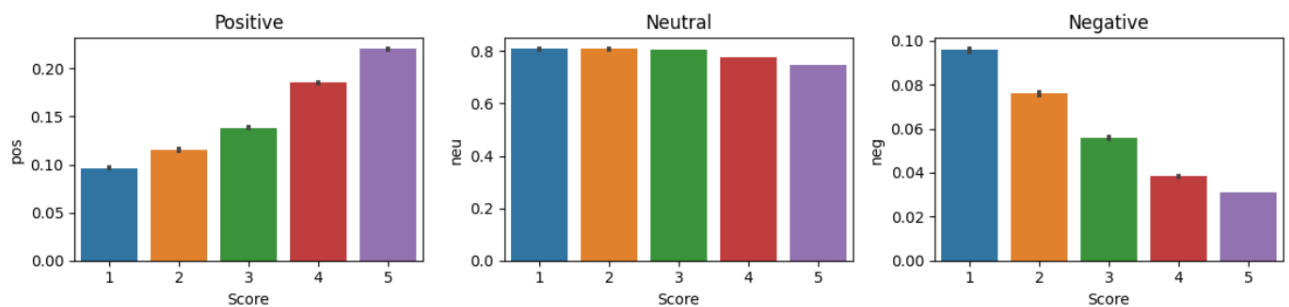
```
: ax = sns.barplot(data=vaders, x='Score', y='compound')
  ax.set_title('Compund Score by Amazon Star Review')
  plt.show()
```





8. Plot of negative, neutral and positive comments.

```
[35]: fig, axs = plt.subplots(1, 3, figsize=(12, 3))
sns.barplot(data=vaders, x='Score', y='pos', ax=axs[0])
sns.barplot(data=vaders, x='Score', y='neu', ax=axs[1])
sns.barplot(data=vaders, x='Score', y='neg', ax=axs[2])
axs[0].set_title('Positive')
axs[1].set_title('Neutral')
axs[2].set_title('Negative')
plt.tight_layout()
plt.show()
```



## Conclusion:

Thus, we have successfully built a cognitive computing application on text data using Python.

