Name : Aryan Deepak Saraf
Div : D20B
Roll No. : 49

AI&DS2 Experiment 05

Aim : To build a Cognitive Analytics for personalization of Customer service application / Insurance / Healthcare Application / Smarter Cities / Government etc.

Theory :

Introduction

Cognitive Analytics is a process of analyzing large amounts of data by combining Artificial Intelligence (AI), Machine Learning (ML), and Natural Language Processing (NLP). The aim is to extract useful knowledge that can help in personalization of services. Unlike normal analytics that just show numbers, cognitive analytics can understand the meaning, emotions, and patterns hidden in the data.

In customer service, healthcare, smarter cities, and government applications, personalization is very important. By analyzing how people feel or what they say in their feedback, a system can adjust its behavior to make services more relevant and user-friendly.

Key Concepts Used in This Experiment

1. Sentiment Analysis
    - A technique in Natural Language Processing used to detect whether a piece of text expresses a positive, negative, or neutral emotion.
    - Helps to understand user satisfaction and opinions.

2. VADER (Valence Aware Dictionary and sEntiment Reasoner)
    - A lexicon-based sentiment analysis tool.
    - Works well for short texts like feedback, reviews, and comments.

3. TextBlob
    - A Python library that provides simple NLP functions.
    - Used for calculating sentiment polarity (positive, neutral, negative).

4. Transformers (BERT-based models)
    - Advanced deep learning models for NLP.
    - Provide more accurate sentiment classification by understanding the context of the sentence.

5. WordCloud Visualization
    - A technique to show the most frequent words in feedback data.
    - Helps in quickly identifying common topics or issues raised by users.

6. Cognitive Analytics for Personalization

- By combining these tools, the system can analyze user feedback and give personalized insights.
- Example: If feedback is mostly negative, the system suggests improvements. If positive, it recommends maintaining the same features or adding enhancements.

Implementation in Smart Gardening Assistant

In this experiment, we extended our Smart Gardening Assistant to include Cognitive Analytics for personalization.

- A dataset of user feedback/comments about the assistant was collected.
- The text data was cleaned and preprocessed (lowercasing, removing special characters, removing short words).
- Sentiment Analysis was performed using:
  - VADER – for quick lexicon-based analysis.
  - TextBlob – for polarity-based scoring.
  - Transformer pipeline – for advanced, context-aware analysis.
- A WordCloud was generated to visualize the most common words in the feedback.
- The system then provided personalized insights such as:
  - If majority feedback is negative → improve advice and accuracy.
  - If majority feedback is positive → continue and expand features.
  - If mixed → balance improvements and monitor continuously.

This makes the assistant more user-centric by adapting its behavior based on real user experiences.

Applications of Cognitive Analytics

1. Customer Service – Understanding customer feedback to improve services.
2. Insurance – Analyzing client reviews and complaints for better claim handling.
3. Healthcare – Monitoring patient feedback on treatments for better personalization.
4. Smarter Cities – Analyzing citizen opinions for city planning and services.
5. Government – Collecting public sentiment to make policy decisions.
6. Smart Gardening (Our Case) – Improving gardening assistant services by analyzing gardener/user feedback.

Code :

```
import pandas as pd
import matplotlib.pyplot as plt
import nltk
import re
```

```
# Download VADER lexicon
nltk.download("vader_lexicon")
from nltk.sentiment.vader import SentimentIntensityAnalyzer
```

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

```
df = pd.read_csv("gardening_feedback_dataset.csv")
print("Gardening Feedback Data (First 10 rows):")
df.head(10)
```

```
Gardening Feedback Data (First 10 rows):
```

	User	Feedback
0	User1	I am dissatisfied with the performance.
1	User2	The experience is okay, can be better.
2	User3	I watered as suggested but the plant looks worse.
3	User4	The assistant is making gardening so easy and ...
4	User5	Advice is not accurate for my garden.
5	User6	Gardening tips are somewhat useful.
6	User7	The soil condition is great and healthy.
7	User8	The assistant helps me save time and my plants...
8	User9	Good advice on watering schedule.
9	User10	Good advice on watering schedule.

```
def clean_text(text):
    text = text.lower()                                  # lowercase
    text = re.sub(r"[^a-z\s]", "", text)                 # remove special chars &
numbers
    text = " ".join([w for w in text.split() if len(w)>2]) # remove short words
    return text
```

```
df["Cleaned_Feedback"] = df["Feedback"].apply(clean_text)
```

```
sia = SentimentIntensityAnalyzer()
df["VADER Score"] = df["Cleaned_Feedback"].apply(lambda x:
sia.polarity_scores(x)["compound"])
df["VADER Sentiment"] = df["VADER Score"].apply(lambda s: "Positive" if s>0
else ("Neutral" if s==0 else "Negative"))
```

```
from textblob import TextBlob
df["TextBlob Score"] = df["Cleaned_Feedback"].apply(lambda x:
TextBlob(x).sentiment.polarity)
df["TextBlob Sentiment"] = df["TextBlob Score"].apply(lambda s: "Positive" if
s>0 else ("Neutral" if s==0 else "Negative"))
```

```
from transformers import pipeline
sentiment_pipeline = pipeline("sentiment-analysis")
```

```
No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and
revision 714eb0f (https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is not recommended.
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
```

(https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(

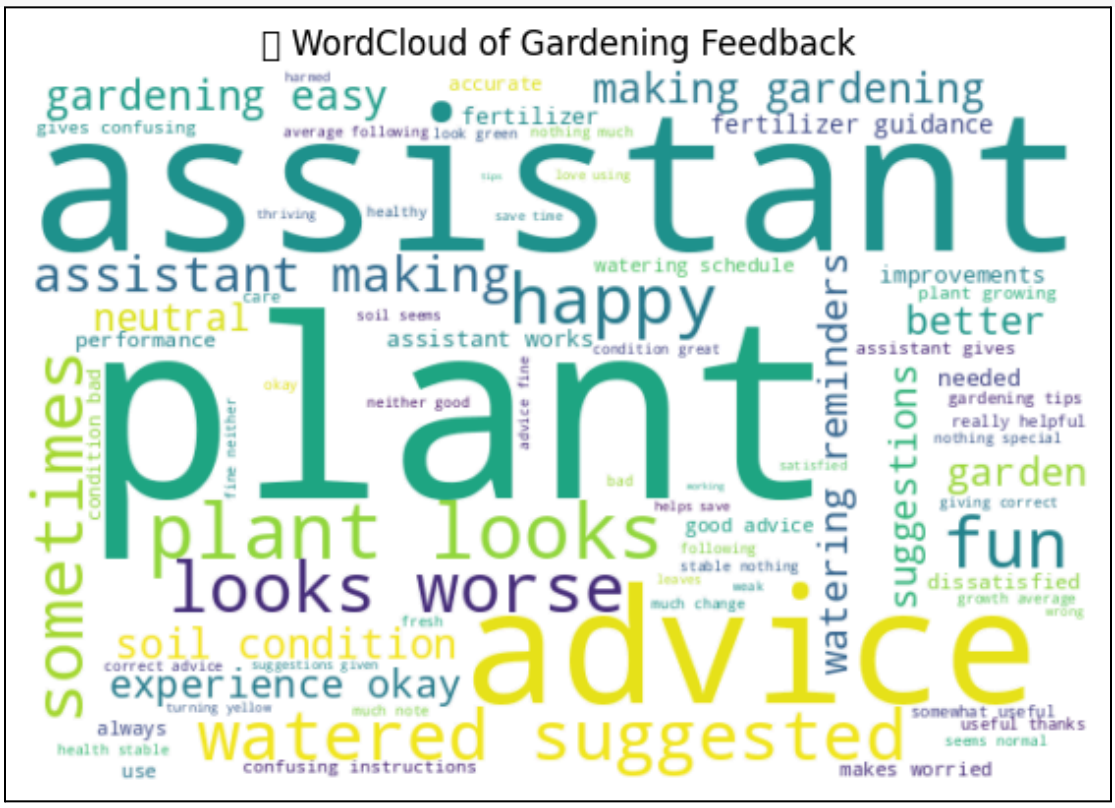
 config.json: 100%
  629/629 [00:00<00:00, 11.5kB/s]
 model.safetensors: 100%
  268M/268M [00:07<00:00, 38.6MB/s]
 tokenizer_config.json: 100%
  48.0/48.0 [00:00<00:00, 1.53kB/s]
 vocab.txt:
  232k/? [00:00<00:00, 3.71MB/s]
Device set to use cpu

```
df["Transformer Result"] = df["Feedback"].apply(lambda x:
sentiment_pipeline(x)[0]["label"])
```

```
from wordcloud import WordCloud

all_text = " ".join(df["Cleaned_Feedback"])
wordcloud = WordCloud(width=600, height=400,
background_color="white").generate(all_text)

plt.figure(figsize=(7,5))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("WordCloud of Gardening Feedback")
plt.show()
```



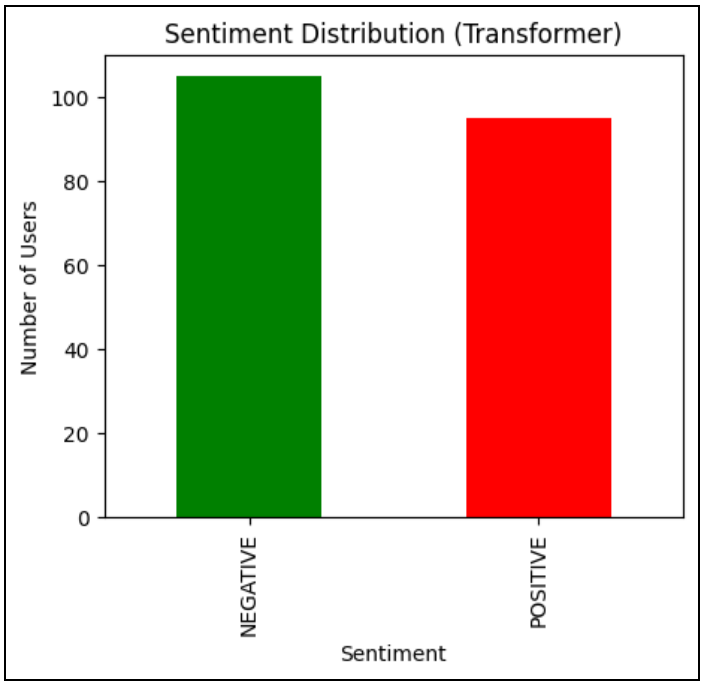
```
print("\nSentiment Analysis Results (First 10 rows):")
df[["User","Feedback","VADER Sentiment","TextBlob Sentiment","Transformer Result"]].head(10)
```

```
Sentiment Analysis Results (First 10 rows):
```

	User	Feedback	VADER Sentiment	TextBlob Sentiment	Transformer Result
0	User1	I am dissatisfied with the performance.	Negative	Neutral	NEGATIVE
1	User2	The experience is okay, can be better.	Positive	Positive	POSITIVE
2	User3	I watered as suggested but the plant looks worse.	Negative	Negative	NEGATIVE
3	User4	The assistant is making gardening so easy and ...	Positive	Positive	POSITIVE
4	User5	Advice is not accurate for my garden.	Neutral	Negative	NEGATIVE
5	User6	Gardening tips are somewhat useful.	Positive	Positive	POSITIVE
6	User7	The soil condition is great and healthy.	Positive	Positive	POSITIVE
7	User8	The assistant helps me save time and my plants...	Positive	Neutral	POSITIVE
8	User9	Good advice on watering schedule.	Positive	Positive	POSITIVE
9	User10	Good advice on watering schedule.	Positive	Positive	POSITIVE

```
# Distribution (using Transformer as final)
sentiment_counts = df["Transformer Result"].value_counts()

plt.figure(figsize=(5,4))
sentiment_counts.plot(kind="bar", color=["green","red","gray"])
plt.title("Sentiment Distribution (Transformer)")
plt.xlabel("Sentiment")
plt.ylabel("Number of Users")
plt.show()
```




```
print("\nPersonalized Insights (Based on Transformer):")
if "NEGATIVE" in sentiment_counts and sentiment_counts["NEGATIVE"] >
sentiment_counts.get("POSITIVE",0):
    print("Many users gave negative feedback. The assistant should improve
advice and accuracy.")
elif "POSITIVE" in sentiment_counts and sentiment_counts["POSITIVE"] >
sentiment_counts.get("NEGATIVE",0):
    print("Most users are satisfied. Keep up the good work and add more
gardening features!")
else:
    print("Mixed feedback received. Balance improvements and continue
monitoring.")
```

```
Personalized Insights (Based on Transformer):
Many users gave negative feedback. The assistant should improve advice and
accuracy.
```

Conclusion : This experiment demonstrated how Cognitive Analytics can be applied to personalize services by analyzing user sentiment and feedback. By using VADER, TextBlob, and Transformer models, the system can understand user emotions with increasing levels of accuracy. Visualization with WordCloud further helps in identifying common patterns. In the Smart Gardening Assistant, this approach allows the system to learn from user feedback and improve its recommendations, making it more adaptive, personalized, and intelligent.

49_Prac_05