Name: **GANESH SANJAY PANDHRE**

Roll Nos : **41** | Div: **D20B**

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

Experiment No: 5

Theory:

Introduction to Cognitive Analytics

Cognitive Analytics is a technique where artificial intelligence (AI) is used to understand data in a way similar to human reasoning. It combines natural language processing, machine learning, and pattern recognition to draw insights from unstructured data like text, speech, or images. Unlike traditional analytics, cognitive analytics does not just give results but also learns context and adapts over time.

Need for Personalization

Personalization means delivering services that are tailored to the specific needs of users. In customer service or smart city applications, every user may have different problems or preferences. By analyzing their behavior, feedback, or complaints, the system can adapt responses and provide relevant recommendations. This improves user satisfaction and efficiency.

Approach Used in the Experiment

In this experiment, resident feedback from an apartment community is used as the dataset. The Apartment Assistant is enhanced with **Cognitive Analytics** to:

- Analyze the **sentiment** of feedback (positive, negative, neutral).
- Identify the **main topics** of feedback (water, maintenance, parking, garden, security, noise, etc.).
- Generate **personalized suggestions** for residents and management based on detected issues.

Two sentiment analysis methods were used:

1. **TextBlob** → a lexicon-based method that measures polarity (positive/negative/neutral).
2. **Transformer-based model (DistilBERT)** → a pre-trained deep learning model that provides advanced sentiment classification.

Additionally, **keyword-based topic classification** was included to link feedback with specific services (e.g., water supply issues trigger water alerts, security feedback improves safety communication).

Visualization and Insights

To better understand the community feedback, the following were used:

- **Sentiment Distribution Graphs** → showing how much feedback is positive, negative, or neutral.
- **WordClouds** → highlighting the most common words in positive and negative feedback, helping identify recurring issues or appreciated services.

Expansion of Apartment Assistant

In previous experiments, the Apartment Assistant mainly focused on answering queries or handling image-based detection. In this experiment, it was **expanded with Cognitive Analytics** so that it can now:

- Learn from **resident opinions** and **adapt** responses.
- Prioritize urgent issues (like water or maintenance complaints).
- Promote positive activities (like events, garden use, community gatherings).
- Provide **personalized communication** to each resident based on their feedback history.

Applications of This Experiment

The methods applied here can also be used in:

- **Customer Service** → analyzing reviews or complaints to improve support.
- **Healthcare** → tracking patient feedback for better care.
- **Insurance** → understanding customer needs to provide personalized plans.
- **Smarter Cities** → analyzing citizen complaints for better governance.
- **Government Applications** → improving public grievance redressal systems.

```
#Install & Import Libraries
!pip install nltk textblob transformers torch wordcloud

import pandas as pd
import numpy as np
import re
import nltk
from textblob import TextBlob
from transformers import pipeline
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

Show hidden output

```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
# Load Dataset
# Upload CSV if running on Colab
from google.colab import files
uploaded = files.upload()
```

Choose Files apartment_…dataset.csv

- **apartment_feedback_dataset.csv**(text/csv) - 49856 bytes, last modified: 8/31/2025 - 100% done

```
Saving apartment_feedback_dataset.csv to apartment_feedback_dataset.csv
```

```
# Read dataset
df = pd.read_csv("apartment_feedback_dataset.csv")
print("Apartment Resident Feedback (First 10 rows):\n")
df.head(10)
```

```
Apartment Resident Feedback (First 10 rows):
```

	Resident	Feedback	Expected_Sentiment
0	Flat101	Security guards are polite and helpful.	Positive
1	Flat101	Garbage collection is irregular and unhygienic.	Negative
2	Flat102	Gym equipment is outdated and needs replacement.	Negative
3	Flat102	I love the garden area, it is well maintained.	Positive
4	Flat103	I am happy with the cleanliness and security i...	Positive
5	Flat103	The staff is very cooperative and supportive.	Positive
6	Flat103	Library is very peaceful and well-maintained.	Positive
7	Flat104	The lift is working smoothly and maintenance s...	Positive
8	Flat104	Electricity supply is stable and uninterrupted.	Positive
9	Flat104	The water supply is very irregular and causes ...	Negative

Next steps: Generate code with df View recommended plots New interactive sheet

```
# Data Cleaning
df['Cleaned_Feedback'] = df['Feedback'].apply(lambda x: re.sub("[^a-zA-Z ]", "", x).lower())
```

```
# Sentiment Analysis with TextBlob
df['Polarity'] = df['Cleaned_Feedback'].apply(lambda x: TextBlob(x).sentiment.polarity)
```

```
def classify_tb(score):
    if score > 0: return "Positive"
    elif score == 0: return "Neutral"
    else: return "Negative"
```

```
df['TextBlob_Sentiment'] = df['Polarity'].apply(classify_tb)
```

```
print("\n🔹 Sample Results using TextBlob:\n")
df[['Resident','Feedback','TextBlob_Sentiment']].head(10)
```

```
🔹 Sample Results using TextBlob:
```

	Resident	Feedback	TextBlob_Sentiment
0	Flat101	Security guards are polite and helpful.	Neutral
1	Flat101	Garbage collection is irregular and unhygienic.	Neutral
2	Flat102	Gym equipment is outdated and needs replacement.	Negative
3	Flat102	I love the garden area, it is well maintained.	Positive
4	Flat103	I am happy with the cleanliness and security i...	Positive
5	Flat103	The staff is very cooperative and supportive.	Positive
6	Flat103	Library is very peaceful and well-maintained.	Positive
7	Flat104	The lift is working smoothly and maintenance s...	Positive
8	Flat104	Electricity supply is stable and uninterrupted.	Neutral
9	Flat104	The water supply is very irregular and causes ...	Positive

```python
# Sentiment Analysis with Transformer (DistilBERT)
classifier = pipeline("sentiment-analysis")

df['Transformer_Result'] = df['Feedback'].apply(lambda x: classifier(x)[0]['label'])
df['Transformer_Score'] = df['Feedback'].apply(lambda x: classifier(x)[0]['score'])

print("\n 🔹 Sample Results using Transformer:\n")
df[['Resident','Feedback','Transformer_Result','Transformer_Score']].head(10)
```

```
No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and
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://huggingfac
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 datas
  warnings.warn(
config.json: 100%                                   629/629 [00:00<00:00, 20.6kB/s]
model.safetensors: 100%                             268M/268M [00:08<00:00, 19.5MB/s]
tokenizer_config.json: 100%                         48.0/48.0 [00:00<00:00, 1.88kB/s]
vocab.txt:     232k/? [00:00<00:00, 3.94MB/s]
Device set to use cpu

 ◆ Sample Results using Transformer:
```

	Resident	Feedback	Transformer_Result	Transformer_Score
0	Flat101	Security guards are polite and helpful.	POSITIVE	0.999672
1	Flat101	Garbage collection is irregular and unhygienic.	NEGATIVE	0.999595
2	Flat102	Gym equipment is outdated and needs replacement.	NEGATIVE	0.999731
3	Flat102	I love the garden area, it is well maintained.	POSITIVE	0.999872
4	Flat103	I am happy with the cleanliness and security i...	POSITIVE	0.999864
5	Flat103	The staff is very cooperative and supportive.	POSITIVE	0.999820
6	Flat103	Library is very peaceful and well-maintained.	POSITIVE	0.999851
7	Flat104	The lift is working smoothly and maintenance s...	POSITIVE	0.999732
8	Flat104	Electricity supply is stable and uninterrupted.	POSITIVE	0.990212
9	Flat104	The water supply is very irregular and causes ...	NEGATIVE	0.999221

```python
# Personalization: Topic-based Recommendations (Keyword Matching)
def topic_from_feedback(text):
    topics = {
        "water": "Provide water supply updates & reminders.",
        "maintenance": "Schedule faster maintenance support.",
        "parking": "Suggest optimized parking management.",
        "event": "Promote upcoming community events.",
        "festival": "Invite to next festival celebration.",
        "garden": "Encourage use of garden and green areas.",
        "noise": "Raise awareness on noise control policies.",
        "security": "Highlight security measures and alerts.",
        "lift": "Ensure regular lift maintenance.",
        "garbage": "Improve garbage collection scheduling.",
        "gym": "Upgrade gym equipment for residents.",
        "library": "Promote library as a peaceful space."
    }
    for k,v in topics.items():
        if k in text: return v
    return "General service feedback logged."
```

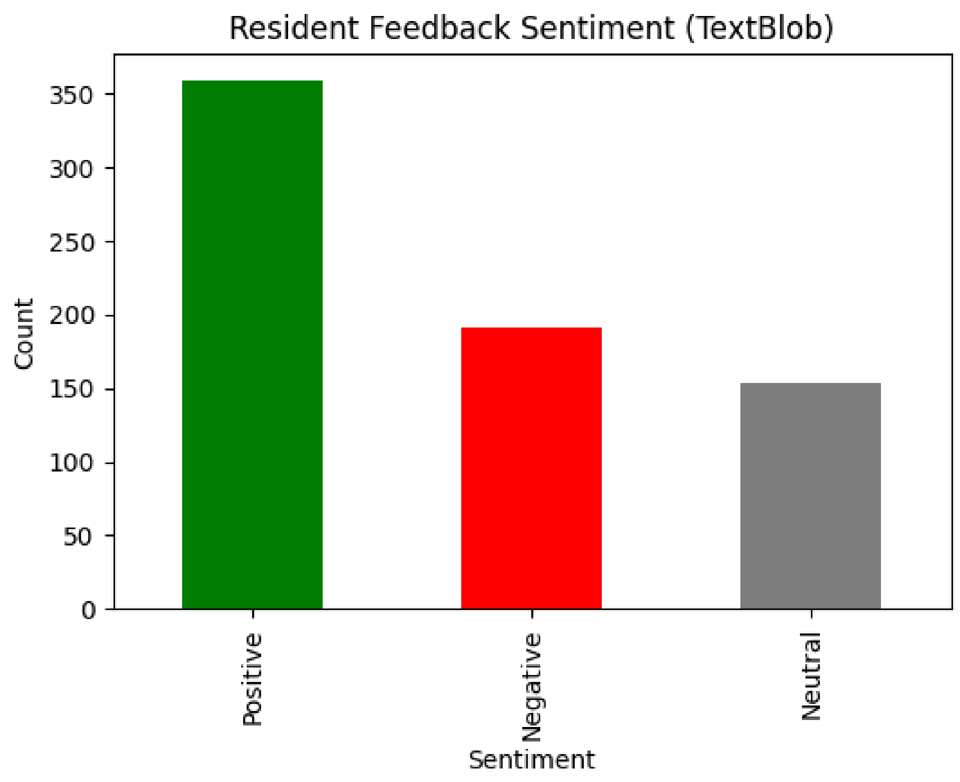
```
df['Personalization'] = df['Cleaned_Feedback'].apply(topic_from_feedback)
```

```
print("\n 🔹 Personalized Suggestions Based on Feedback Topics:\n")
df[['Resident','Feedback','TextBlob_Sentiment','Personalization']].head(10)
```

```
🔹 Personalized Suggestions Based on Feedback Topics:
```

	Resident	Feedback	TextBlob_Sentiment	Personalization
0	Flat101	Security guards are polite and helpful.	Neutral	Highlight security measures and alerts.
1	Flat101	Garbage collection is irregular and unhygienic.	Neutral	Improve garbage collection scheduling.
2	Flat102	Gym equipment is outdated and needs replacement.	Negative	Upgrade gym equipment for residents.
3	Flat102	I love the garden area, it is well maintained.	Positive	Encourage use of garden and green areas.
4	Flat103	I am happy with the cleanliness and security i...	Positive	Highlight security measures and alerts.
5	Flat103	The staff is very cooperative and supportive.	Positive	General service feedback logged.
6	Flat103	Library is very peaceful and well-	Positive	Promote library as a peaceful

```
# Visualization Sentiment Distribution
plt.figure(figsize=(6,4))
df['TextBlob_Sentiment'].value_counts().plot(kind='bar', color=['green','red','gray'], title="Resident
plt.xlabel("Sentiment")
plt.ylabel("Count")
plt.show()
```

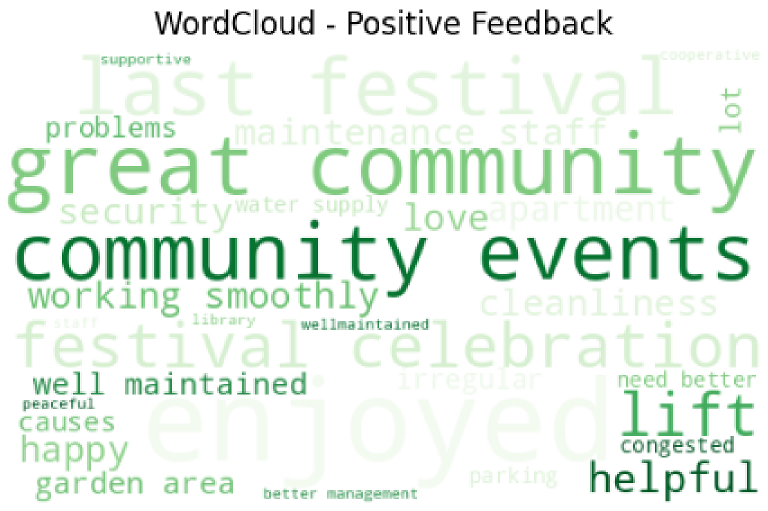


```
# Visualization : WordCloud for Positive & Negative Sentiments
positive_text = " ".join(df[df['TextBlob_Sentiment']=="Positive"]['Cleaned_Feedback'])
negative_text = " ".join(df[df['TextBlob_Sentiment']=="Negative"]['Cleaned_Feedback'])
```

```
wc_pos = WordCloud(width=500, height=300, background_color="white", colormap="Greens").generate(positi
wc_neg = WordCloud(width=500, height=300, background_color="white", colormap="Reds").generate(negative_
```

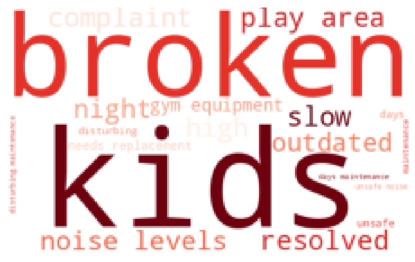
```
plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
plt.imshow(wc_pos)
plt.axis("off")
plt.title("WordCloud - Positive Feedback")
```

```
Text(0.5, 1.0, 'WordCloud - Positive Feedback')
```



```
plt.subplot(1,2,2)
plt.imshow(wc_neg)
plt.axis("off")
plt.title("WordCloud - Negative Feedback")
plt.show()
```

WordCloud - Negative Feedback



Conclusion

This experiment demonstrated how **Cognitive Analytics enables personalization** in an Apartment Assistant. By combining sentiment analysis, keyword detection, and visualization, the system can better understand resident needs, prioritize issues, and adapt its services. This makes the assistant more intelligent, context-aware, and resident-friendly, marking a significant expansion from earlier versions.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.