Name : Naikwadi Yash Shivdas

AI&DS II 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 branch of artificial intelligence that combines data analysis and machine learning to mimic human thought processes in understanding complex data. In customer service applications—for sectors like insurance, healthcare, smart cities, and government—cognitive analytics helps personalize service delivery by understanding customer sentiments and preferences.

Sentiment Analysis in Personalization

Sentiment analysis is a key tool of cognitive analytics that interprets customer opinions from textual data such as feedback, reviews, or social media comments. It classifies opinions into categories like positive, negative, or neutral sentiment. This polarity detection allows businesses to gauge customer satisfaction in real-time.

Three popular approaches for sentiment analysis include:

- Rule-based and Lexicon approaches, such as VADER (Valence Aware Dictionary and sEntiment Reasoner), that use predefined dictionaries and heuristics.
- Statistical and machine learning methods, including TextBlob, which use linguistic features to compute polarity.
- Deep learning and transformer-based models that leverage large pretrained language models (e.g., BERT) for more nuanced understanding.

Data Preprocessing

Effective sentiment analysis requires preprocessing steps such as:

- Cleaning text data by removing punctuations, numbers, and special characters.
- Lowercasing for consistency.
- Tokenization to split text into words or symbols.
- Removing stop words to focus on meaningful terms.

Cognitive Analytics for Personalization

By integrating sentiment analysis results into customer profiles, service systems can:

- Tailor responses and recommendations dynamically based on user sentiment.
- Detect dissatisfaction early and provide targeted support.
- Highlight positive feedback to reinforce good service.
- Inform strategic decisions for service improvement.

For example, a food delivery service can analyze customer feedback to identify common complaints like delayed delivery or poor packaging and offer personalized discounts or faster responses to affected customers.

Visualization and Interpretation

Visualization techniques such as Word Clouds help extract and display the most frequent terms from customer comments, revealing key topics and concerns visually, which aids decision-making.

Summary

This experiment demonstrates how cognitive analytics using multiple sentiment analysis techniques—including lexicon-based, statistical, and transformer models—enables effective personalization for customer service applications. It underscores the importance of analyzing customer feedback intelligently to improve service quality and user satisfaction in various domains.

CODE:

Libraries Import and Setup

```
!pip install -q nltk textblob transformers wordcloud
!python -m textblob.download_corpora
!pip install -q torch
```

```
import pandas as pd
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from textblob import TextBlob
from transformers import pipeline
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import re
```

Download required NLTK data

```
nltk.download("vader_lexicon")
```

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

True
```

Initialize sentiment analyzers

```
vader = SentimentIntensityAnalyzer()
transformer_sentiment = pipeline('sentiment-analysis')
```

Read CSV file into dataframe

```
df = pd.read_csv('/content/feedback.csv')
print(df.head())
```

```
   User                                    Comment
0  User1      The pizza was delicious and arrived on time.
```

```
1  User2       Food arrived cold and the delivery was late.
2  User3    Amazing service, very satisfied with the order.
3  User4  The food quality was poor and the packaging wa...
4  User5    Fast delivery and tasty food, will order again.
```

Function to clean text for WordCloud

```
def clean_text(text):
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    text = text.lower()
    return text
```

Apply VADER sentiment analysis

```
def vader_sentiment(text):
    score = vader.polarity_scores(text)['compound']
    if score > 0.05:
        return "Positive"
    elif score < -0.05:
        return "Negative"
    else:
        return "Neutral"
```

Apply TextBlob sentiment polarity

```
def textblob_sentiment(text):
    polarity = TextBlob(text).sentiment.polarity
    if polarity > 0:
        return "Positive"
    elif polarity == 0:
        return "Neutral"
    else:
        return "Negative"
```

Apply Transformer sentiment classification

```
def transformer_sentiment_label(text):
    result = transformer_sentiment(text)[0]
    label = result['label']
    return label
```

Apply all sentiment analyses

```
df['VADER_Sentiment'] = df['Comment'].apply(vader_sentiment)
df['TextBlob_Sentiment'] = df['Comment'].apply(textblob_sentiment)
df['Transformer_Sentiment'] = df['Comment'].apply(transformer_sentiment_label)
```

Display dataframe with sentiments

```
df
```

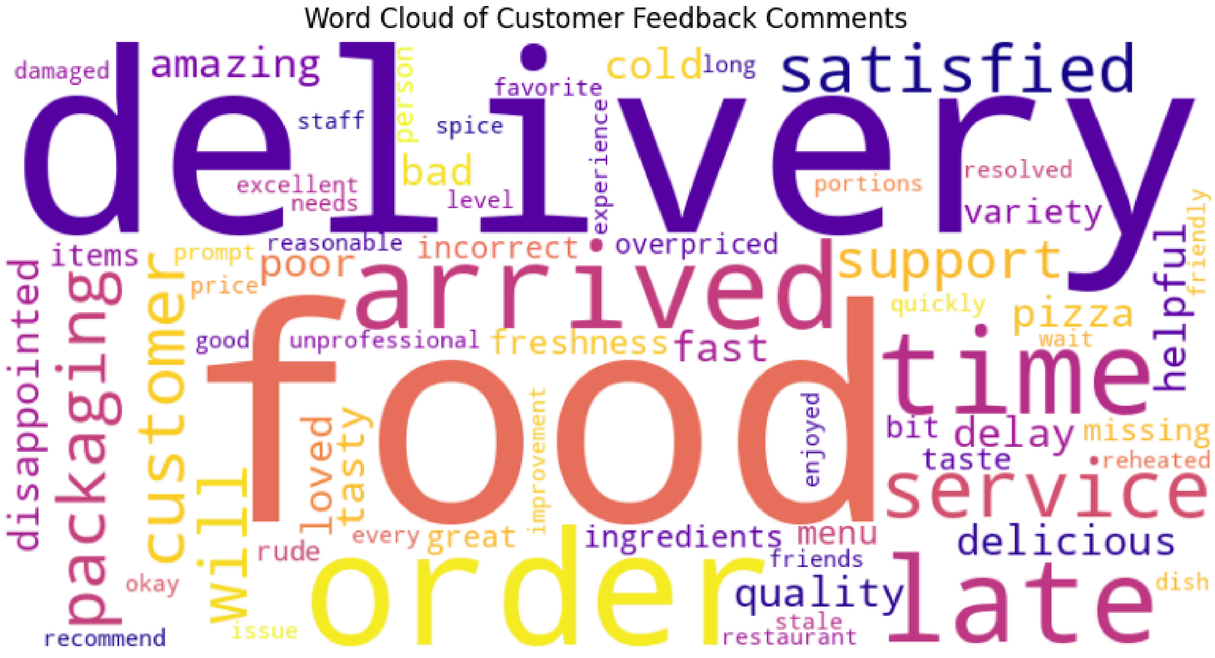
	User	Comment	VADER_Sentiment	TextBlob_Sentiment	Transformer_Sentiment	Personalized_Response
0	User1	The pizza was delicious and arrived on time.	Positive	Positive	POSITIVE	Thank you for the positive feedback! We're gla...

1	User2	Food arrived cold and the delivery was late.	Neutral	Negative	NEGATIVE	Thanks for your feedback. We'll keep striving ...
2	User3	Amazing service, very satisfied with the order.	Positive	Positive	POSITIVE	Thank you for the positive feedback! We're gla...
3	User4	The food quality was poor and the packaging wa...	Negative	Negative	NEGATIVE	We're sorry for the inconvenience caused. Our ...
4	User5	Fast delivery and tasty food, will order again.	Neutral	Positive	POSITIVE	Thanks for your feedback. We'll keep striving ...
5	User6	Disappointed with the delay, but customer supp...	Positive	Negative	POSITIVE	Thank you for the positive feedback! We're gla...

Generate and display WordCloud

```
all_text = ' '.join(df['Comment'].apply(clean_text))
wordcloud = WordCloud(width=800, height=400, background_color='white',
colormap='plasma').generate(all_text)
```

```
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Customer Feedback Comments')
plt.show()
```



Personalization example based on overall sentiment distribution

```
sentiment_counts = df['VADER_Sentiment'].value_counts()
print("Sentiment distribution (VADER):")
print(sentiment_counts)
```

```
Sentiment distribution (VADER):
VADER_Sentiment
```

```
Positive    12
Neutral      4
Negative     4
Name: count, dtype: int64
```

```python
def personalized_response(sentiment):
    if sentiment == "Positive":
        return "Thank you for the positive feedback! We're glad you enjoyed our service."
    elif sentiment == "Neutral":
        return "Thanks for your feedback. We'll keep striving to improve."
    else:
        return "We're sorry for the inconvenience caused. Our team will work to resolve the issues."
```

```python
# Example: Generate personalized responses for each comment
df['Personalized_Response'] = df['VADER_Sentiment'].apply(personalized_response)

print("\nSample personalized responses:")
df[['Comment', 'VADER_Sentiment', 'Personalized_Response']]
```

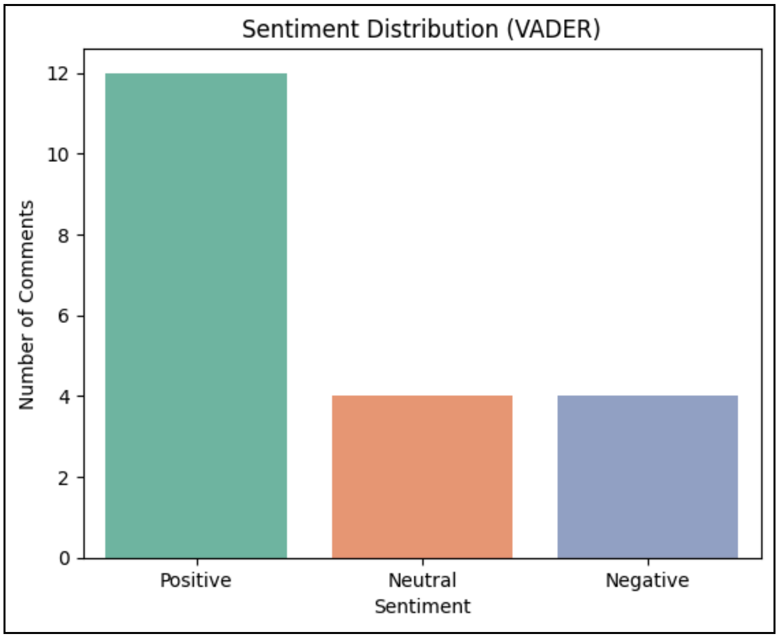
Sample personalized responses:

	Comment	VADER_Sentiment	Personalized_Response
0	The pizza was delicious and arrived on time.	Positive	Thank you for the positive feedback! We're gla...
1	Food arrived cold and the delivery was late.	Neutral	Thanks for your feedback. We'll keep striving ...
2	Amazing service, very satisfied with the order.	Positive	Thank you for the positive feedback! We're gla...
3	The food quality was poor and the packaging wa...	Negative	We're sorry for the inconvenience caused. Our ...
4	Fast delivery and tasty food, will order again.	Neutral	Thanks for your feedback. We'll keep striving ...
5	Disappointed with the delay, but customer supp...	Positive	Thank you for the positive feedback! We're gla...

Sentiment Distribution Bar Chart

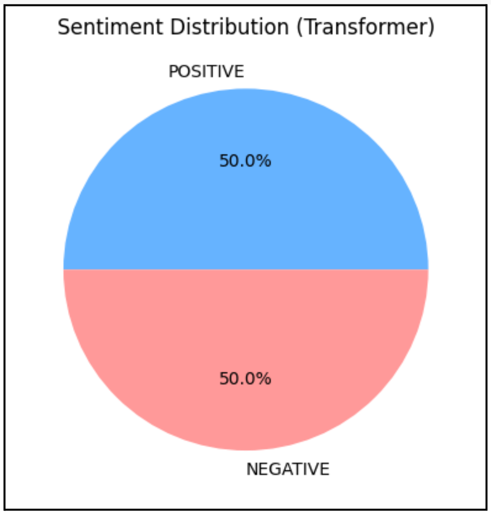
```python
import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(x='VADER_Sentiment', data=df, palette='Set2')
plt.title('Sentiment Distribution (VADER)')
plt.xlabel('Sentiment')
plt.ylabel('Number of Comments')
plt.show()
```



Sentiment Comparison Pie Chart

```
sentiment_counts = df['Transformer_Sentiment'].value_counts()
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%',
colors=['#66b3ff','#ff9999','#99ff99'])
plt.title('Sentiment Distribution (Transformer)')
plt.show()
```



CONCLUSION:

This experiment successfully demonstrated the application of cognitive analytics techniques, such as sentiment analysis using VADER, TextBlob, and transformer models, to analyze customer feedback and personalize customer service. By understanding customer sentiments and preferences, the system can provide meaningful and targeted responses, enhancing user satisfaction and improving service quality. Cognitive analytics thus plays a vital role in making customer service more intelligent, responsive, and personalized across various domains.

39_colab_05