



```
[12]: import openai

[2]: openai.api_key = "pase you key here"

•[20]: def sentiment_analysis(text):
import openai

# Set up the API key (avoid using your actual key here; load it securely from environment variables if possible)
openai.api_key = "Paste Your API_key here"

messages = [
    {"role": "system", "content": "You are trained to analyze and detect the sentiment of given text. If you're unsure of an answer, you can say 'not s"},
    {"role": "user", "content": f"Analyze the following text and determine if the sentiment is positive or negative. Return answer in a single word as ."}
]

response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=messages,
    temperature=0
)

organized_response = response.choices[0].message['content'].strip()
return organized_response

prompt = "I love Pakistan"
res = sentiment_analysis(prompt)
print(f"{prompt} => Sentiment analysis is: {res}")

I love Pakistan => Sentiment analysis is: positive
```

😊Code Explanation😊

1. **Import Libraries:**
 - `import openai`: Imports the OpenAI library for accessing GPT models.
2. **Set API Key:**
 - `openai.api_key = "Paste Your API Key Here"`: Sets up the API key required to use OpenAI's API. (Replace "Paste Your API Key Here" with the actual key).
3. **Define sentiment_analysis Function:**
 - `def sentiment_analysis(text)::` Defines a function named `sentiment_analysis` that takes text as an input for sentiment analysis.
4. **Define Messages for Chat Completion:**
 - Creates a list of messages, `messages`, that includes:
 - A **system message** instructing the AI to analyze sentiment.
 - A **user message** asking the AI to analyze the input text (`text`) and provide the result as either "positive" or "negative."
5. **API Request:**
 - `openai.ChatCompletion.create(...)`: Calls the GPT-3.5-turbo model to analyze the sentiment based on the provided messages. `temperature=0` ensures deterministic responses (consistent output).
6. **Process and Return Output:**
 - Extracts the model's response, trims any extra whitespace, and returns it.
7. **Example Usage:**
 - Calls `sentiment_analysis` with the prompt "I love Pakistan" and prints the result.

Output: The printed output shows that the sentiment analysis for "I love Pakistan" is positive.