

Problema Elegido: La ineficiencia en la atención al cliente en empresas de comercio electrónico.

Una atención al cliente deficiente puede llevar a la pérdida de clientes y una mala reputación de la empresa. Los clientes esperan respuestas rápidas y precisas a sus consultas, lo cual puede ser difícil de lograr sin la ayuda de tecnologías avanzadas. Mejorar la atención al cliente puede aumentar la satisfacción del cliente y, en última instancia, las ventas.

Solución Propuesta: Utilizar prompts específicos para generar respuestas automáticas y eficaces a las consultas más comunes de los clientes utilizando IA.

Se utilizarán modelos de lenguaje IA, como los ofrecidos por OpenAI, para generar respuestas personalizadas y precisas a las consultas de los clientes. Esto incluye tanto modelos texto-texto (para responder preguntas frecuentes) como texto-imagen (para proporcionar visuales de productos o soluciones gráficas).

Justificación de la Viabilidad del Proyecto

Viabilidad Técnica: Disponemos de acceso a la API de OpenAI y un equipo de desarrolladores con experiencia en integración de IA. Además, las herramientas necesarias para probar y ajustar los prompts están fácilmente disponibles. Justificación de Elecciones: Hemos elegido utilizar la IA para mejorar la atención al cliente debido a su capacidad para manejar grandes volúmenes de consultas de manera eficiente y efectiva. Esta tecnología permite proporcionar respuestas rápidas y precisas, mejorando la experiencia del cliente y liberando tiempo para que los agentes humanos puedan manejar consultas más complejas. Respuestas Inmediatas: Las respuestas automáticas brindan soluciones rápidas a las consultas de los clientes, sin demoras. Reducción de Tareas Repetitivas: La IA automatiza respuestas estándar, liberando a los agentes humanos de tareas monótonas.

Caso 1

```
In [83]: import openai
openai.api_key = ""

In [85]: # Contexto y prompt
context = """
Situación: Un cliente pregunta sobre el estado de su pedido usando su número de identificación. Propósito: Informar al cliente sobre el estado actual de su pedido y la
fecha estimada de entrega para mejorar su experiencia y gestionar expectativas.
Información del Pedido:
- Número de Pedido: 378
- Estado: Enviado
- Fecha Estimada de Entrega: 5 de agosto de 2024
"""
prompt = "¿Cuál es el estado de mi pedido con número 378?"

In [ ]: conversation = [
    {"role": "system", "content": context},
    {"role": "user", "content": prompt}
]
response = openai.ChatCompletion.create(
    model='gpt-3.5-turbo',
    messages=conversation,
    max_tokens=500,
    temperature=0.7
)
# Corrected way to access the response content
```

```
message = response['choices'][0]['message']['content']
print("Assistant: {}".format(message))
```

Caso 2:

```
import openai
openai.api_key = ""

context = """ Situación: Un usuario está teniendo problemas para procesar su pago en el sitio web. Propósito: Asistir al usuario en resolver el problema con su pago, ofreciendo alternativas y soporte adicional si es necesario. """
prompt = "Estoy teniendo problemas para procesar mi pago. ¿Qué puedo hacer?"
```

```
conversation = [
    {"role": "system", "content": context},
    {"role": "user", "content": prompt}
]

response = openai.ChatCompletion.create(
    model='gpt-3.5-turbo',
    messages=conversation,
    max_tokens=500,
    temperature=0.7
)

# Extraer y mostrar la respuesta
message = response['choices'][0]['message']['content']
print("Assistant: {}".format(message))
```

prompt_imagen = "Una persona que tiene problemas al intentar realizar un pago en línea, muestra una expresión confusa y mira una pantalla de computadora con un mensaje de pago rechazado".



```
Caso 3:
import openai
openai.api_key = ""

context = """ Situación: Un usuario desea devolver un producto y solicitar un reembolso. Propósito: Guiar al usuario a través del proceso de devolución y reembolso para asegurar una experiencia sin problemas."""
prompt = "¿Cómo puedo devolver un producto y solicitar un reembolso?"

conversation = [
    {"role": "system", "content": context},
    {"role": "user", "content": prompt}
]
response = openai.ChatCompletion.create(
    model='gpt-3.5-turbo',
    messages=conversation,
    max_tokens=500,
    temperature=0.7
)

# Extraer y mostrar la respuesta
message = response['choices'][0]['message']['content']
print("Assistant: {}".format(message))

response_imagen = openai.Image.create (prompt="Crea una ilustración de un cliente que devuelve un producto y recibe un reembolso, mostrando un paquete que se entrega a un mensajero y se reembolsa el dinero ",
n=1,
size="1024x1024")

image_url = response_imagen['data'][0]['url']
print(image url)
```

