Problem Statement:

Many customers shopping on e-commerce platforms face challenges and inconveniences during their shopping experience, such as difficulty in finding products, lack of personalized recommendations, unclear product information, and limited support options. These issues often result in reduced customer satisfaction, increased cart abandonmentrates, and lost sales opportunities for e-commerce businesses. To address these problems, there is a need for an intelligent and efficient chatbot solution in the e commerce industry that can provide seamless assistance, enhance user experience, and drive higher conversion rates.

Objective:

Enhance customer experience,

Increase sales,

Reduce cart abandonment by providing personalized product recommendations, Quick answers to queries,

Seamless shopping assistant.

Design Thinking:

User Scenarios:

When the users are interacting with the chatbot.

The frequent questions asked by the users like forgot password. Conversation help

Feedback of the chatbot experience

Response Configuration:

The response of the chatbot is under three stages of processing o Intent detection – to identify the overall idea of the query.

o Entity detection – to identify the keyword of the query and fetch sub parts of the query.

o Irrelevant detection – to identify the query given by the user is query or not.

o Auto error correction – to auto correct the human errors. (forgot

Platform Integration

To integrate with the other popular platforms similar to us like

o Facebook messenger

To improve the efficiency and performance of the chatbot and identify the user type.

User Experience:

To know the experience by taking the feedback survey after the end of the conversation

5 star-excellent,4star-good,

Creating a ChatBot for e-commerce using AI can enhance customer support, streamline operations, and improve the overall shopping experience.

Here's a step-by-step guide on how to build aChatBot for an e-commerce platform using AI To provide customer support, help with product recommendations, process orders.

Create conversational flows that cater to various scenarios, such as product inquiries,order status, returns, and FAQs. Map out the dialogues and decision trees.

Setting up the FAQs datasets:

{

"questions": [

{

"question": "How can I create an account?",

"answer": "To create an account, click on the 'Sign Up' button on the top right corner of our

website and follow the instructions to complete the registration process."

},

{

"question": "What payment methods do you accept?",

"answer": "We accept major credit cards, debit cards, and PayPal as payment methods for online

orders."

},

{

"question": "How can I track my order?",

"answer": "You can track your order by logging into your account and navigating to the 'Order History'

section. There, you will find the tracking information for your shipment."

},

{
"question": "What is your return policy?",
"answer": "Our return policy allows you to return products within 30 days of purchase for a full refund, provided they are in their original condition and packaging. Please refer to our Returns page for detailed instructions."
},
{
"question": "Can I cancel my order?",
"answer": "You can cancel your order if it has not been shipped yet. Please contact our customer support team with your order details, and we will assist you with the cancellation process." },
{
"question": "How long does shipping take?",
"answer": "Shipping times vary depending on the destination and the shipping method chosen. Standard shipping usually takes 3-5 business days, while express shipping can take 1-2 business days." },
{
"question": "Do you offer international shipping?",
"answer": "Yes, we offer international shipping to select countries. The availability and shipping costs will be calculated during the checkout process based on your location."
},
{
"question": "What should I do if my package is lost or damaged?",
"answer": "If your package is lost or damaged during transit, please contact our customer support team immediately. We will initiate an investigation and take the necessary steps to resolve the issue."
},
{

"question": "Can I change my shipping address after placing an order?",
"answer": "If you need to change your shipping address, please contact our customer support team as soon as possible. We will do our best to update the address if the order has not been shipped yet."
},

{
"question": "How can I contact customer support?",
"answer": "You can contact our customer support team by phone at [phone number] or by email at [email address]. Our team is available [working hours] to assist you with any inquiries or issues you may have."
},
{
"question": "Do you offer gift wrapping services?",
"answer": "Yes, we offer gift wrapping services for an additional fee. During the checkout process, you can select the option to add gift wrapping to your order."
},
{
"question": "What is your price matching policy?",
"answer": "We have a price matching policy where we will match the price of an identical product found on a competitor's website. Please contact our customer support team with the details of the product and the competitor's offer."
},
{
"question": "Can I order by phone?",
"answer": "Unfortunately, we do not accept orders over the phone. Please place your order through our website for a smooth and secure transaction."
},

{

"question": "Are my personal and payment details secure?",

"answer": "Yes, we take the security of your personal and payment details seriously. We use industry-standard encryption and follow strict security protocols to ensure your information is protected."

},

{

"question": "What is your price adjustment policy?",

"answer": "If a product you purchased goes on sale within 7 days of your purchase, we offer a one-time price adjustment. Please contact our customer support team with your order details to request the adjustment."

},

{

"question": "Do you have a loyalty program?",

"answer": "Yes, we have a loyalty program where you can earn points for every purchase. These points can be redeemed for discounts on future orders. Please visit our website to learn more and join the program."

},

{

"question": "Can I order without creating an account?",

"answer": "Yes, you can place an order as a guest without creating an account. However, creating an account offers benefits such as order tracking and easier future purchases."

},

{

"question": "Do you offer bulk or wholesale discounts?",

"answer": "Yes, we offer bulk or wholesale discounts for certain products. Please contact our

customer support team or visit our Wholesale page for more information and to discuss your specific requirements."

},

{

"question": "Can I change or cancel an item in my order?",

"answer": "If you need to change or cancel an item in your order, please contact our customer support team as soon as possible. We will assist you with the necessary steps."

},

{

"question": "How can I leave a product review?",

"answer": "To leave a product review, navigate to the product page on our website and click on the 'Write a Review' button. You can share your feedback and rating based on your experience with the product."

},

]

}

These are the some examples of frequently asked questions samples.

Ensure your chatbot has an intuitive and user-friendly design. Users should easily understand how to interact with it.

Create conversational flows that cater to various scenarios, such as product inquiries, order status, returns, and FAQs. Map out the dialogues and decision trees.

Model training and machine learning algorithm:

Let's begin by installing the chatterbot library. For creating chatbot also need to install chatterbot corpus. Corpus - literal meaning is a collection of words. This contains a corpus of data that is included in the chatterbot module. Each corpus is nothing but a prototype of different input statements and their

responses. These corpus are used by bots to train themselves. The most recommended method for installing chatterbot and chatterbot_corpus is by using pip.

Installation commands for terminal:

pip install chatterbot:

pip install chatterbot_corpus;

Connect Database

Storage Adapters allows you to connect to a particular storage unit or network. For using a storage adapter, we need to specify it. We will position the storage adapter by assigning it to the import path of the storage we want to use. Here we are using SQL Storage Adapter, which permits chatbot to connect to databases in SQL. By using the database parameter, we will create a new SQLite Database.

```
bot = ChatBot(
'Buddy',
storage_adapter='chatterbot.storage.SQLStorageAdapter',
database_uri='sqlite:///database.sqlite3'
)
```

Matching the keyword

You can also position the logical adapter with a chatbot object. As the name implies, Logical Adapter regulates the logic behind the chatterbot, i.e., it picks responses for any input provided to it. This parameter contains a list of logical

operators. Chatterbot allows us to use a number of logical Adapters. When more than one logical adapter is put to use, the chatbot will calculate the

confidence level, and the response with the highest calculated confidence will be returned as output. Here we have used two logical adapters: BestMatch and TimeLogicAdapter.

```
# Create object of ChatBot class with Logic Adapter
bot = ChatBot(
'Buddy',
logic_adapters=[
'chatterbot.logic.BestMatch',
'chatterbot.logic.TimeLogicAdapter'],
)
```

ChatBot Testing

The last step of this tutorial is to test the chatterbot's conversational skills. For testing its responses, we will call the get_responses() method of Chatbot instance.

```
# Get a response to the input text 'I would like to book a flight.'
response = bot.get_response('I have a complaint.')
print("Bot Response:", response)
```

output:

Bot Response: Please elaborate, your concern

In this example, we will create a simple chatbot that can answer general questions about an e-commerce store. We'll use ChatterBot for natural language processing.

Coding

```
from chatterbot import ChatBot

from chatterbot.trainers import ChatterBotCorpusTrainer


# Create a chatbot instance

chatbot = ChatBot('EcommerceBot')


# Create a new trainer for the chatbot

trainer = ChatterBotCorpusTrainer(chatbot)
# Train the chatbot on the English language

trainer.train("chatterbot.corpus.english")


while True:

user_input = input("You: ")


if user_input.lower() == 'quit':

break


response = chatbot.get_response(user_input)

print("EcommerceBot:", response)
```

Product Enquires

```
def get_product_info(product_name):

if product_name in products:

return response_prompts["product"].format(

product_name=product_name,

product_price=products[product_name]["price"],

product_description=products[product_name]["description"
```

```
] )

else:

return response_prompts["not_found"]


def chatbot_response(user_input):

user_input = user_input.lower()


if any(greet in user_input for greet in

greetings): return response_prompts["greet"]

elif "bye" in user_input:
return response_prompts["goodbye"]

elif "price" in user_input:

return "You can ask about product prices by specifying the product

name." else:

for product_name in products:

if product_name in user_input:

return get_product_info(product_name)

return "I'm sorry, I didn't understand that. Please ask another question

or specify a product."
```

End

We will create a while loop for our chatbot to run in. When statements are passed in the loop, we will get an appropriate response for it, as we have already entered data into our database. If we get "Bye" or "bye" statement from the user, we can put an end to the loop and stop the program.

```
name=input("Enter Your Name: ")

print("Welcome to the Bot Service! Let me know how can I help

you?") while True:
```

```
request=input(name+':')

if request=='Bye' or request =='bye':

print('Bot: Bye')

break

else:

response=bot.get_response(request)

print('Bot:',response)
```

output:

```
Enter Your Name: John
Welcome to the Bot Service! Let me know how can I help you?
John: Hi
Bot: Hello
John: I need your assistance regarding my order
Bot: Please, Provide me with your order id
John: I have a complaint.
Bot: Please elaborate, your concern
John: How long it will take to receive an order?
Bot: An order takes 3-5 Business days to get delivered.
John: Okay Thanks
Bot: No Problem! Have a Good Day!
John: Bye
Bot: Bye
```