ARTIFICIAL INTELLIGENCE

NAME: HANISHKHA. K

TITTLE: CREATE A CHATBOT IN PYTHON

PROBLEM DEFINITION:

Satisfying shopping experiences are the key to building a loyal following of paying customers. If people don't have immediate answers to their most pressing questions, you could lose a sale and a potential lifetime customer with real value.

But what happens if your customer support team is overwhelmed by the volume of incoming questions? What if you don't have enough people power to respond to those enquiries in real time

In situations where your inbound traffic is higher than can reasonably be managed by human support staff, an AI chatbot is one of your best resources to handle the higher volume. **Chatbots can be programmed to respond to FAQs in real time and, thanks to their machine learning capabilities, they can identify the intent behind those questions to strike up more meaningful, informative, and satisfying conversations with shoppers.** This helps automate the customer support process and ensure shoppers receive the support necessary to satisfy their needs.

chatbot work as;

There are three templates of chat-based solutions that enable real time engagement with buyers with the potential to relieve overburdened support teams. However, not all are made equal.

- **Live chat software**
- **Rule-based chatbots**

- **AI chatbots**

Live chat software

Live chat is a manual way to have direct engagement with shoppers as they browse through your website, basically the equivalent of Facebook Messenger. According to Super Office, 79% of customers prefer using live chat over hopping on a phone call with a support agent as it provides immediate responses to pressing questions.

With live chat, you gain specific insight into your buyers' needs and provide direct responses to those enquiries. Using those responses, you can guide shoppers to the most valuable point of conversion that will give them the items they desire while increasing revenue for your business.

Rule-based chatbots

Rule-based chatbots are examples of the types of bots that people have interacted with for years across many websites. They use a predefined set of rules, primarily built upon if/then statements that inform the bots of how to automatically respond to FAQs in place of a human.

The benefit of rule-based chatbots is that they're easy to build and train to interact with shoppers. The drawback is that because they're programmed using specific rule-based responses, they're limited in how they can engage in helpful conversation with shoppers.

AI chatbots

As opposed to rule-based chatbots, AI chatbots can make proactive recommendations to shoppers, triggered by browsing behaviors. The chatbots use natural language processing (NLP) and machine learning technology to react to the buyer intent of shopping behaviours.

Companies use AI chatbots to not only automate customer support, but they also help optimise conversion rates, improve customer experiences, and contribute to direct revenue. In fact, AI chatbots have helped generate a 60% increase in revenue for e-commerce businesses.

A chatbot is a computer program that can simulate a conversation or chat with a user in natural language through messaging applications, website, or mobile applications and interact with users. Chatbots are developed and became so popular due to the increased use of smart devices and IT technology.

Types of chatbots

a. Baseline chatbot:

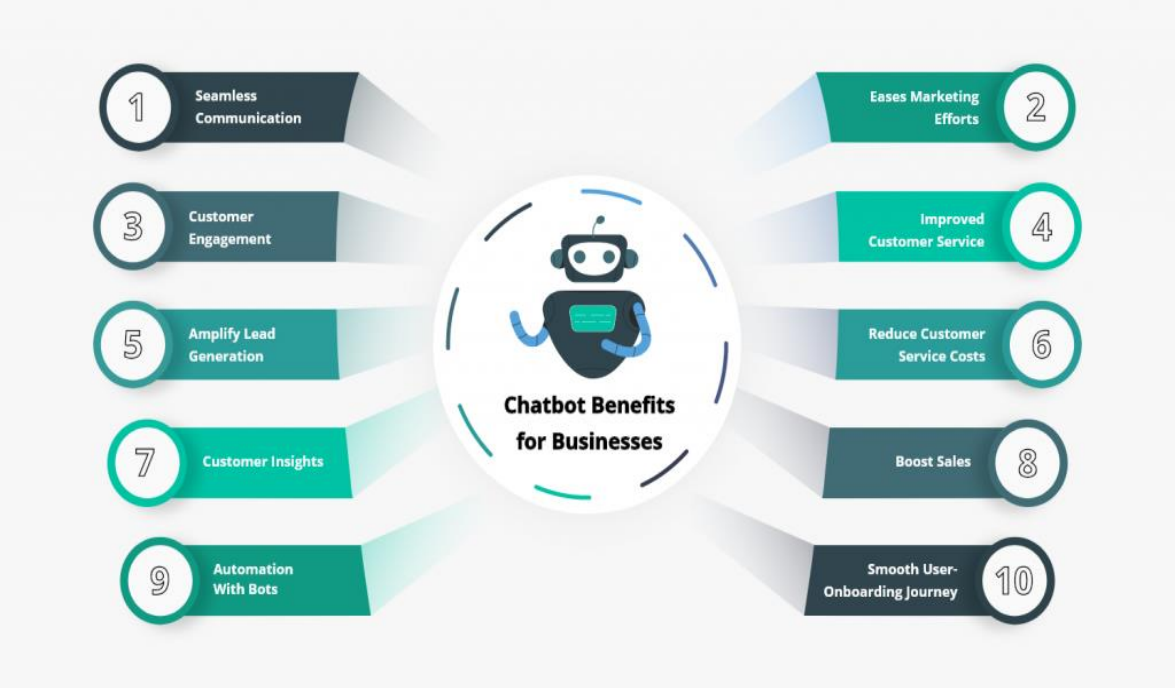
It is a chatbot that is based on a database and uses if / then logic to create a conversation flow and that takes a lot of time to ensure the understanding of the question and the answer needed.

b. AI chatbot:

This type of chatbot is more complex than base-line but it is more interactive and personalized and needs big data training to be impressive if the problem is matched to their capabilities.

c. Hybrid Model:

A hybrid approach mixes the Base-line & AI chatbot to make it smart and his behavior more expected by depending on database and Ai algorithm to work together. Chatbot Platforms alternatives:



DESIGN THINKING

Empathize

To empathize, you need to do our research. Consult with experts (even if that's your customer support team), talk to our users, and immerse our self in the environment. If we plan to make a Facebook Messenger bot (to add to the more than 100k already available), spend time speaking with our users via the app. Find out what their needs are. Seek out pain points. Set aside our own assumptions and *learn*. This is a perfect example of the old adage "An ounce of prevention is worth a pound of cure."

Define

our goal here is to define our problem in a human-centered (not business-centered) way. Don't frame it as "We need to build a chatbot to lower our response time for support tickets." Look at it more like "People don't want to spend a bunch of time looking for a solution or a help doc." This approach presents a larger, more open-ended problem, and that'll help out when we get into the next step, Ideate.

Ideate

our goal here isn't to come up with The One Best Idea. It's to generate as many ideas as possible. Think freely (or "outside the box," if we must). There are plenty of approaches to this part of the process like brainstorming,mindmapping, SCAMPER, Worst Possible.

In a field as new and unexplored as chatbot design, this stage of design thinking could lead we almost anywhere. Good news: That's okay. This isn't the time to get stuck on voice and tone or types of prompts. This is the time when we start to think about human vs. automated responses, where the answers are going to come from, and how deep we want to go to save your users time and potential frustration.

Prototype

This is an experimental phase, and the goal is to identify the best possible solutions for the problems identified during the first three stages.

our design team will produce some low-fi, scaled-down versions of your chatbot (and its guiding logic) in an effort to find what works. Every idea that survived the transition into Prototyping will either be rejected (which is what will happen to most of them) or accepted, revised, and improved.

Test

The Testing stage is where your designers, your researchers, and possibly even some of your users come together to test the more polished prototypes that were the results of your prototyping.

While this is the fifth step that's been outlined, it doesn't necessarily have to be the last. As a matter of fact, most of the time the information and feedback we gather in the Test stage leads us to re-define our problem or to better empathize with our users.

HOW DO MAKE A CHATBOT IN PYTHON

To build a chatbot in Python, we must import all the necessary packages and initialize the variables we use in our chatbot project. Also, remember that when working with text data, we need to perform data preprocessing on our dataset before designing an ML model.

This is where tokenizing helps with text data – it helps fragment the large text dataset into smaller, readable chunks (like words). Once that is done, you can also go for lemmatization that transforms a word into its lemma form. Then it creates a pickle file to store the python objects that are used for predicting the responses of the bot. Another vital part of the chatbot development process is creating the training and testing datasets. Now that we've covered the basics of chatbot development in Python, let's dive deeper into the actual process! It will help you understand how to create a chatbot.

1. Prepare the Dependencies

The first step in creating a chatbot in Python with the Chatterbot library is to install the library in your system. It is best if you create and use a new Python virtual environment for the installation.

```
pip install --upgrade chatterbot_corpus
pip install --upgrade chatterbot
```

2. Import Classes

Importing classes is the second step in the Python chatbot creation process. we need to do is import two classes – ChatBot from chatterbot and List Trainer from chatterbot. Trainers.

```
from chatterbot.trainers import ListTrainer
```

3. Create and Train the Chatbot

This is the third step on creating chatbot in python. The chatbot we are creating will be an instance of the class "ChatBot." After creating a new ChatterBot instance, we can train the bot to improve its performance. Training ensures that the bot has enough knowledge to get started with specific responses to specific inputs.

```
                                'chatterbot.logic.BestMatch'])
```

Here, the argument (that corresponds to the parameter name) represents the name of your Python chatbot. If you wish to disable the bot's ability to learn after the training, can include the "read_only=True" command. The command "logic_adapters" denotes the list of adapters used to train the chatbot.

While the "chatterbot.logic.MathematicalEvaluation" helps the bot to solve math problems, the "chatterbot.logic.BestMatch" helps it to choose the best match from the list of responses already provided.

Since you have to provide a list of responses, you can do it by specifying the lists of strings that can be later used to train your Python chatbot, and find the best match for each query.

```
    list_trainer.train(item)
```

4. Communicate with the Python Chatbot

To interact with our Python chatbot, we can use the .get_response() function.However, it is essential to understand that the chatbot using python might not know how to answer all our questions. Since its knowledge and training is still very limited, we must give it time and provide more training data to train it further.

5. Train your Python Chatbot with a Corpus of Data

In this last step of how to make a chatbot in Python, for training our python chatbot even further, we can use an existing corpus of data. Here's an example of how to train our Python chatbot with a corpus of data provided by the bot itself:

```
corpus_trainer.train('chatterbot.corpus.english')
```

Develpment

A chatbot is one of the simple ways to transport data from a computer without having to think for proper keywords to look up in a search or browse several web pages to collect information; users can easily type their query in natural language and retrieve information