**Problem statement**

1. **Eliminates the added costs to meet global customer demands**
2. **Automates repeat customer support enquiries**
3. **Ends sales activity only taking place during working hours**
4. **Reduces abandoned carts**
5. **Gives customers an accessible channel to find answers to their questions**

**Libraries used and the integration of NLP techniques**

**To create an NLP chatbot, define its scope and capabilities, collect and preprocess a dataset, train an NLP model, integrate it with a messaging platform, develop a user interface, and test and refine the chatbot based on feedback.**

**How to chatbot interactive with users**

**Chatbots interact with users by using natural language processing (NLP) and machine learning (ML) technologies to understand and respond to user input. They are designed to mimic human conversation and can be integrated into various platforms, such as websites, mobile apps, messaging apps, and more.**

**Innovation techniques**

**Map confusion rate (CR) of your virtual assistant. ...**

**Leverage NLP to enhance the understanding of your bot on colloquial say. ...**

**Know Your Audience & Hyper-personalize. ...**

**Frame Empathetic Responses using advanced sentiment analysis.**

**Complication explanation**

**Pip install chatterbot**

**This code installs the Python package called “chatterbot” using the pip package manager.**

**• Pip is a package installer for Python that allows you to easily install and manage third-party libraries and packages.**

**• The “chatterbot” package is a Python library that makes it easy to generate automated responses to a user’s input, making it useful for building chatbots and conversational agents.**

**• By running this code, the “chatterbot” package will be downloaded and installed on the user’s system, making it available for use in their Python code.**

pip install chatterbot\_corpus

**This code installs the Python package called “chatterbot\_corpus” using the pip package manager.**

**• The “chatterbot\_corpus” package contains pre-built training data for the ChatterBot library, which is a Python library for creating chatbots.**

**• The “pip install” command is used to install Python packages from the PyPI (Python Package Index) repository.**

**• When this command is executed, pip will download the “chatterbot\_corpus” package and its dependencies (if any) from PyPI and install them on the local machine.**

**!pip install chatterbot**

**This code installs the Python package called “chatterbot” using the pip package manager.**

**• The exclamation mark at the beginning of the line indicates that this is a command to be executed in the command line interface (CLI) rather than in Python code.**

**• The pip package manager is used to install and manage Python packages, and the “install” command is used to install a package.**

**• The “chatterbot” package is a Python library that enables developers to build chatbots and conversational agents.**

**!pip install chatterbot\_corpus**

**OpenAI**

**This code installs the Python package called “chatterbot\_corpus” using the pip package manager.**

**• The exclamation mark at the beginning of the line indicates that this is a command to be executed in the command line interface (CLI) rather than in Python code.**

**• The “pip” command is used to install packages in Python, and “chatterbot\_corpus” is the name of the package being installed.**

**# Create object of ChatBot class**

**Bot = ChatBot(‘Buddy’)**

**This code creates an object of the ChatBot class and assigns it to the variable bot.**

**• The ChatBot class is likely defined in a library or module that has been imported into the current Python script.**

**• The argument ‘Buddy’ is passed to the ChatBot constructor, which is likely used to set the name of the chatbot.**

**[nltk\_data] Downloading package averaged\_perceptron\_tagger to**

**[nltk\_data] /root/nltk\_data...**

**[nltk\_data] Unzipping taggers/averaged\_perceptron\_tagger.zip.**

**[nltk\_data] Downloading package stopwords to /root/nltk\_data...**

**[nltk\_data] Unzipping corpora/stopwords.zip.**

**[nltk\_data] Downloading package wordnet to /root/nltk\_data...**

**[nltk\_data] Unzipping corpora/wordnet.zip.**

**Bot = ChatBot(**

**‘Buddy’,**

**Storage\_adapter=’chatterbot.storage.SQLStorageAdapter’,**

**Database\_uri=’sqlite:///database.sqlite3’**

**)**

**Bot = ChatBot(**

**‘Buddy’,**

**Logic\_adapters=[**

**‘chatterbot.logic.BestMatch’,**

**‘chatterbot.logic.TimeLogicAdapter’],**

**This code creates an instance of a chatbot using the ChatBot class from the chatterbot library.**

**• The chatbot is given the name “Buddy”.**

**• The logic\_adapters parameter is used to specify the logic adapters that the chatbot will use to generate responses.**

**• In this case, the chatbot will use two logic adapters: BestMatch and TimeLogicAdapter.**

**• The BestMatch adapter uses a combination of cosine similarity and Levenshtein distance to find the closest matching response to a given input.**

**• The TimeLogicAdapter adapter allows the chatbot to respond to questions about the current time.**

**• Overall, this code creates a chatbot instance with two logic adapters that will be used to generate responses to user input**

**# Inport ListTrainer**

**From chatterbot.trainers import ListTrainer**

**Trainer = ListTrainer(bot)**

**Trainer.train([**

**‘Hi’,**

**‘Hello’,**

**‘I need your assistance regarding my order’,**

**‘Please, Provide me with your order id’,**

**‘I have a complaint.’,**

**‘Please elaborate, your concern’,**

**‘How long it will take to receive an order ?’,**

**‘An order takes 3-5 Business days to get delivered.’,**

**‘Okay Thanks’,**

**‘No Problem! Have a Good Day!’**

**])**