Here's a basic example of a chatbot using Python and NLP libraries. This is a simplified version and might need modifications to fit your specific requirements.

Required Libraries:

- nltk for natural language processing

- sklearn for machine learning

- pandas for data manipulation

Code:

import nltk

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

import pandas as pd

# Sample dataset for training

data = {

"query": ["What is your return policy?", "How do I track my order?", "What is your shipping policy?"],

"response": ["You can return items within 30 days.", "You can track your order using the tracking number.", "We offer free shipping on orders over $50."]

}

df = pd.DataFrame(data)

# Tokenize and remove stopwords

nltk.download('punkt')

nltk.download('stopwords')

stop\_words = set(stopwords.words('english'))

def preprocess\_text(text):

tokens = word\_tokenize(text)

tokens = [t for t in tokens if t.isalpha() and t not in stop\_words]

return ' '.join(tokens)

df['query'] = df['query'].apply(preprocess\_text)

# Vectorize text data

vectorizer = TfidfVectorizer()

X = vectorizer.fit\_transform(df['query'])

y = df['response']

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train a Naive Bayes classifier

clf = MultinomialNB()

clf.fit(X\_train, y\_train)

# Define a function to respond to user queries

def respond\_to\_query(query):

query = preprocess\_text(query)

query\_vector = vectorizer.transform([query])

response = clf.predict(query\_vector)

return response[0]

# Test the chatbot

user\_query = "What is your return policy?"

response = respond\_to\_query(user\_query)

print(response)

This code trains a simple Naive Bayes classifier to respond to user queries based on a sample dataset. You can modify and extend this code to fit your specific requirements.

Note: This is a basic example and might not work perfectly for all scenarios. You may need to add more data, fine-tune the model, and handle edge cases to improve the chatbot's performance.