```
# Sentiment Chatbot (Stops when you type 'exit')

import nltk
import re
import string
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression

# Download required NLTK data
nltk.download('punkt', quiet=True)
nltk.download('stopwords', quiet=True)

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

stop_words = set(stopwords.words('english'))

# ----------------------------
# Expanded Training Data
# ----------------------------
texts = [
    # Positive
    "I am very happy",
    "This is amazing",
    "I love this product",
    "You are doing great",
```

```
    "I feel fantastic today",
    "That was wonderful",
    "This makes me smile",
    "I am excited about this",
    "Everything is awesome",
    "I am proud of you",

    # Negative
    "I am very sad",
    "This is terrible",
    "I hate this",
    "I feel disappointed",
    "This is the worst",
    "I am upset",
    "That was horrible",
    "I feel angry",
    "This makes me frustrated",
    "I am not happy"
]

labels = [1]*10 + [0]*10

# ----------------------------
# Preprocessing Function
# ----------------------------
def preprocess(text):
```

```
    text = text.lower()
    text = re.sub(r'\d+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    tokens = word_tokenize(text)
    tokens = [w for w in tokens if w not in stop_words]
    return " ".join(tokens)

processed_texts = [preprocess(t) for t in texts]

# ----------------------------
# TF-IDF + Model Training
# ----------------------------
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(processed_texts)

model = LogisticRegression()
model.fit(X, labels)

print("🤖 Chatbot Ready! Type 'exit' to stop.\n")

# ----------------------------
# Interactive Chat Loop
# ----------------------------
while True:
    user_input = input("You: ")
```

```
if user_input.lower() == "exit":
    print("Chatbot: Goodbye! 👋")
    break

processed_input = preprocess(user_input)
input_vector = vectorizer.transform([processed_input])
prediction = model.predict(input_vector)[0]

if prediction == 1:
    print("Chatbot: 😊 You sound positive!")
else:
    print("Chatbot: 😔 You seem upset.")
```