Python Code

```
import re
import nltk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
import numpy as np

# Download necessary NLTK resources (if not already downloaded)
nltk.download('punkt')

# Sample product-related keywords for Toutche
product_keywords = ["bicycle", "electric", "heileo", "battery", "range",
"features"]

# Basic text preprocessing function
def preprocess_text(text):
    # Convert text to lowercase
    text = text.lower()
    # Remove punctuation using regular expressions
    text = re.sub(r'[^\w\s]', '', text)
    # Tokenize the text
    tokens = nltk.word_tokenize(text)
    return tokens

# Function to identify keywords related to Toutche's products
def identify_keywords(text):
    tokens = preprocess_text(text)
    found_keywords = [word for word in tokens if word in
product_keywords]
    return found_keywords

# Sample user queries for intent classification
training_data = [
    ("Tell me about the Heileo electric bicycle", "product_info"),
    ("What is the price of the battery?", "pricing"),
    ("I need technical support for my bicycle", "technical_support"),
```

```
    ("How long does the battery last?", "product_info"),
    ("Can you help me with my electric bike?", "technical_support"),
]

# Tokenizing and preparing training data for intent classification
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform([text for text, label in training_data])
y_train = np.array([label for text, label in training_data])

# Train a simple logistic regression model for intent classification
model = LogisticRegression()
model.fit(X_train, y_train)

# Function to classify user queries
def classify_intent(user_input):
    X_test = vectorizer.transform([user_input])
    predicted_intent = model.predict(X_test)[0]
    return predicted_intent

# Example usage
if __name__ == "__main__":
    # Preprocess and identify keywords
    sample_text = "I want to know more about the Heileo electric bicycle
features."
    keywords = identify_keywords(sample_text)
    print("Identified Keywords:", keywords)

    # Classify user intent
    user_query = "What is the price of the Heileo electric bike?"
    intent = classify_intent(user_query)
    print("Classified Intent:", intent)
```

Code Explanation

1. **Imports**: We import necessary libraries including `re` for regular expressions, `nltk` for natural language processing tasks, and `scikit-learn` for machine learning.

2. **Download NLTK Resources**: We ensure that the necessary NLTK resources are available for tokenization.

3. **Product Keywords**: A list of keywords related to Toutche's products is created, which will be used to identify relevant terms in user inputs.

4. **Preprocessing Function**: The `preprocess_text` function performs several tasks:

    - Converts input text to lowercase to ensure uniformity.
    - Removes punctuation using a regular expression.
    - Tokenizes the cleaned text using NLTK's `word_tokenize`.

5. **Keyword Identification Function**: The `identify_keywords` function processes the input text to find and return any keywords related to Toutche's products.

6. **Training Data for Intent Classification**: We prepare training data for a simple intent classification model. Each user query is labeled with its corresponding intent.

7. **Vectorization**: The training text is vectorized using `CountVectorizer`, which converts the text into a format suitable for the logistic regression model.

8. **Model Training**: A logistic regression model is trained using the vectorized training data.

9. **Intent Classification Function**: The `classify_intent` function takes user input, vectorizes it, and predicts the intent using the trained model.

10. **Example Usage**: The script demonstrates how to identify keywords from a sample text and classify the intent of a user query.