```
import pandas as pd
data = pd.read_csv('/content/drugs_side_effects_drugs_com.csv')
```

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
nltk.download('punkt')
nltk.download('stopwords')
def preprocess_text(text):
    """Preprocess text by tokenizing, lowercasing, and removing stopwords."""
    if isinstance(text, str):
        tokens = word_tokenize(text.lower())
        tokens = [word for word in tokens if word.isalnum() and word not in stopwords.words('english')]
        return ' '.join(tokens)
    else:
        return ''
data['processed_side_effects'] = data['side_effects'].apply(preprocess_text)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
import spacy
nlp = spacy.load("en_core_web_sm")
def extract_entities(text):
    """Extract entities from text using NER."""
    doc = nlp(text)
    entities = [(ent.text, ent.label_) for ent in doc.ents]
    return entities
sample_description = data['processed_side_effects'].iloc[0]
entities = extract_entities(sample_description)
print(entities)
```

```
[('several weeks', 'DATE')]
```

```
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
llm_model = SentenceTransformer('all-MiniLM-L6-v2')
embeddings = llm_model.encode(data['processed_side_effects'].tolist())
def find_similar_drugs_with_llm(drug_description):
    """Find similar drugs based on cosine similarity of embeddings from LLM."""
    description_embedding = llm_model.encode([drug_description])
    similarities = cosine_similarity(description_embedding, embeddings).flatten()
    return similarities.argsort()[-5:][::-1]
similar_drugs_indices = find_similar_drugs_with_llm(data['processed_side_effects'].iloc[0])
print(data.iloc[similar_drugs_indices][['drug_name', 'side_effects']])
```

```
/usr/local/lib/python3.10/dist-packages/sentence_transformers/cross_encoder/CrossEncoder.py:13: TqdmExperimentalWarning: Using `tqdm
  from tqdm.autonotebook import tqdm, trange
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
modules.json: 100%                                                  349/349 [00:00<00:00, 9.70kB/s]
config_sentence_transformers.json: 100%                             116/116 [00:00<00:00, 7.11kB/s]
README.md: 100%                                                     10.7k/10.7k [00:00<00:00, 507kB/s]
sentence_bert_config.json: 100%                                     53.0/53.0 [00:00<00:00, 1.48kB/s]
config.json: 100%                                                   612/612 [00:00<00:00, 19.0kB/s]
model.safetensors: 100%                                             90.9M/90.9M [00:00<00:00, 120MB/s]
tokenizer_config.json: 100%                                         350/350 [00:00<00:00, 21.5kB/s]
vocab.txt: 100%                                                     232k/232k [00:00<00:00, 3.48MB/s]
tokenizer.json: 100%                                                466k/466k [00:00<00:00, 7.33MB/s]
special_tokens_map.json: 100%                                       112/112 [00:00<00:00, 7.00kB/s]
1_Pooling/config.json: 100%                                         190/190 [00:00<00:00, 8.09kB/s]
```

	drug_name	side_effects
0	doxycycline	(hives, difficult breathing, swelling in your ...
2648	procaine penicillin	hives ; difficulty breathing; swelling of your...
622	Geodon	(hives, difficult breathing, swelling in your ...
2813	zonisamide	any form of skin rash , hives ; fever, swollen...
1864	metolazone	hives ; difficult breathing; swelling of your ...

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
y = (data['medical_condition'] == 'Acne').astype(int)
X_train, X_test, y_train, y_test = train_test_split(embeddings, y, test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
print(f'Precision: {precision:.2f}')
print(f'Recall: {recall:.2f}')
```