エラー対策_RAG機能付きアプリをstreamlit_community_cloudにアップロードしたときのエラー対策

yamda2000

エラー対策1

```
import faiss #faiss-cpuのインストールが必要
from langchain_community.vectorstores import FAISS #faiss-cpuのインストールが必要
```

import faiss #faiss-cpuのインストールが必要
from langchain_community.vectorstores import FAISS #faiss-cpuのインストールが必要

```
#db = Chroma.from_documents(docs, embedding=embeddings, persist_directory="./chroma_db") #1回目
db = FAISS.from_documents(docs, embedding=embeddings) #faiss-cpuのインストールが必要
# フォルダがなければ作成
os.makedirs("./FAISS_db", exist_ok=True)
# 保存
db.save_local("./FAISS_db")
```

#db = Chroma.from_documents(docs, embedding=embeddings, persist_directory="./chroma_db") #1回目 streamlit community cloudでエラーがでる。
db = FAISS.from_documents(docs, embedding=embeddings) #faiss-cpuのインストールが必要
# フォルダがなければ作成
os.makedirs("./FAISS_db", exist_ok=True)
# 保存
db.save_local("./FAISS_db")

エラー対策2

```python
# LLMレスポンスのテキストを辞書に変換
product_lines = result[0].page_content.split("\n")
product = {item.split(": ")[0]: item.split(": ")[1] for item in product_lines}

# キー名の前後の空白やBOMを取り除く(例：'id'⇒'id')
product = {k.strip().lstrip("\ufeff"): v for k, v in product.items()}
```


```
# LLMレスポンスのテキストを辞書に変換
product_lines = result[0].page_content.split("¥n")
product = {item.split(": ")[0]: item.split(": ")[1] for item in product_lines}


# キー名の前後の空白やBOMを取り除く(例：'id'⇒'id')
product = {k.strip().lstrip("¥ufeff"): v for k, v in product.items()}
```