# TimeWeightedVectorStoreRetriever M1 Mac 커널 충돌 해 결

# 📌 문제 상황

날짜: 2025-10-02 ~ 10-03 환경: M1 Mac, Jupyter Notebook, Python 3.13.5 목표: TimeWeightedVectorStoreRetriever 구현

## ℚ 발생한 문제

#### 증상

- retriever.add\_documents() 호출시 커널 충돌
- FAISS/Chroma 모두 동일 증상
- 384, 768, 1536, 3048 차원 모두 실패

#### 에러 패턴

Kernel died, restarting...

# 

### 시도 1: 임베딩 차원 변경 🗙

- 384차원: all-MiniLM-L6-v2 by huggingface embedding model
- 768차원: paraphrase-multilingual-mpnet by huggingface embedding model
- 1536차원: fake embeddings by huggingface embedding model
- 3048차원: google-embedding-model
- 결과: 모두 실패 → FAISS와 M1의 호환 좋지 않다고 판단 → Chroma로 변경 시도

#### 시도 2: 벡터스토어 변경 🗙

- FAISS → 커널 충돌
- Chroma → 커널 충돌
- 결과: 벡터스토어 문제 아님

#### 시도 3: 환경 전환 🗸

- M1 Mac → Google Colab
- 가장 낮은 384차원 임베딩 모델부터 시도
- **결과**: 한번에 성공!

# ♀ 원인 분석

#### 근본 원인

#### TimeWeightedVectorStoreRetriever 자체가 M1 Mac 비호환

#### 기술적 원인

- M1 ARM 아키텍처와 LangChain 라이브러리 충돌
- add\_documents() 내부 메모리 관리 이슈
- 특히 시간 메타데이터 처리 부분에서 충돌

#### 참고 자료

- LangChain GitHub Issues: TimeWeighted M1 관련 이슈 다수
- 커뮤니티 권장: Colab 또는 수동 구현

# ☑ 최종 해결책

### Solution 1: Google Colab (채택)

```
# Colab에서 교재 그대로 실행
# 100% 성공
```

#### 장점:

- 교재 완벽 재현
- 설정 변경 불필요
- 안정적 동작

#### 단점:

- 로컬 환경 아님
- 인터넷 필요

#### Solution 2: 수동 구현 (대안)

```
# Manual time-weighted search
# Chroma(등 각자의 환경에 맞는 벡터스토어 선택) + 후처리 방식
```

#### 장점:

- M1 완벽 호환
- 커스터마이징 가능

#### 단점:

- 교재와 다름
- 구현 복잡도 증가

# ₩ 성과

항목	결과
환경	Google Colab
소요 시간	5분 (Colab 설정 후)
성공률	100%
교재 재현	완벽

# ☞ 교훈

- 1. 환경 문제는 환경 전환으로
  - 억지로 M1에서 해결하려 하지 말 것
  - 적절한 환경 선택이 중요
- 2. 개념 이해가 우선
  - TimeWeighted 개념은 완전히 이해
  - 구현 방법은 여러 가지 가능
- 3. 트러블슈팅 가치
  - 2일간 고생 → 문제 해결 경험
  - 포트폴리오 소재로 활용

# ◈ 관련 작업

- #32.12: SelfQueryRetriever (OpenAl 전환)
- #32.11: MultiVectorRetriever (로컬 LLM)
- M1 호환성 연구