Jakie czynniki wpływają na wielkość ryzyka związanego z działalnością na instrumentach pochodnych?

Jako czynniki wpływające na wielkość podejmowanego ryzyka należy rozumieć w szczególności stopy procentowe, kursy wymiany walut, indeksy, ceny na rynku towarowym i kapitałowym (towary i inne instrumenty finansowe) oraz ich zmienność, zmiany zdolności kredytowej klientów, zmiany płynności rynku, możliwość wystąpienia dużych wstrząsów na rynkach oraz ewentualnych zjawisk kryzysowych.

**Obecnie**

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.

**Gemini reranking (ocena po uzyskaniu odpowiednich akapitów)**

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Zawartość wygenerowana przez AI może być niepoprawna.

**Gemini reranking + hybrid search (uzyskanie odpowiednich akapitów z ChromaDB + semantyczny embedding Gemini)**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Zawartość wygenerowana przez AI może być niepoprawna.**

# Reranking step in **process\_query.\_get\_rag\_context**

        reranking\_model = genai.GenerativeModel(RAG\_RERANKING\_EMBEDDING)

        query\_emb = np.array(reranking\_model.embed\_content(content=prompt, task\_type="retrieval\_query")["embedding"])

        passage\_texts = [text for text, \_ in passages\_with\_pages]

        passage\_embs = []

        for passage in passage\_texts:

            resp = reranking\_model.embed\_content(content=passage, task\_type="retrieval\_document")

            passage\_embs.append(np.array(resp["embedding"]))

        scores = cosine\_similarity([query\_emb], passage\_embs)[0]

        reranked = sorted(zip(passages\_with\_pages, scores), key=lambda x: x[1], reverse=True)

        top\_reranked = reranked[:n\_results]

# Hybrid Search in **rag\_use\_chroma\_collections.get\_relevant\_passage**

# Keyword-based ChromaDB retrieval

    results = db.query(query\_texts=[query], n\_results=n\_results)

    passages = results["documents"][0]

    metadatas = results["metadatas"][0]

    chroma\_chunks  = []

    for i, passage in enumerate(passages):

        page\_number = metadatas[i].get("page\_number", "unknown")

        chroma\_chunks.append((passage, page\_number))

    # Embedding-based semantic Gemini retrieval

    try:

        hs\_model = genai.GenerativeModel("embedding-001")

        query\_embedding = np.array(hs\_model.embed\_content(content=query, task\_type="retrieval\_query")["embedding"])

        all\_data = db.get(include=["documents", "metadatas"])

        all\_docs = all\_data["documents"]

        all\_metas = all\_data["metadatas"]

        doc\_page\_pairs = list(zip(all\_docs, all\_metas))

        dense\_texts = [doc[0] for doc in doc\_page\_pairs]

        dense\_embeddings = [np.array(hs\_model.embed\_content(content=txt, task\_type="retrieval\_document")["embedding"])

            for txt in dense\_texts]

        similarities = cosine\_similarity([query\_embedding], dense\_embeddings)[0]

        dense\_ranked = sorted(zip(doc\_page\_pairs, similarities), key=lambda x: x[1], reverse=True)

        gemini\_chunks = [(text, meta.get("page\_number", "unknown")) for (text, meta), \_ in dense\_ranked[:n\_results]]

    except Exception as e:

        log.warning(f"Gemini embedding failed: {e}")

        gemini\_chunks = []

    # Merge results from both retrievals

    seen\_passages = set()

    combined\_passages = []

    for passage, page in chroma\_chunks + gemini\_chunks:

        if passage not in seen\_passages:

            combined\_passages.append((passage, page))

            seen\_passages.add(passage)

1. Rekomendacja P
2. Numpy i sklearn

Co jest przeprowadzane w ramach analizy portfeli, w skład których wchodzą instrumenty pochodne?

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.