Version 1.0: ragtest.py - Foundational PDF RAG System

- **Core Theme:** Command-Line Interface (CLI) based RAG system for PDF documents using Groq LLM.
- **Key Features:**
    - Configuration loaded from config.ini.
    - PDF text/table extraction (pdfplumber), chunking, Sentence Transformer embeddings, FAISS indexing.
    - Query pipeline: retrieval, context aggregation, Groq LLM for answers.

Version 2.0: ragtest_c.py - Streamlit UI Integration

- **Core Theme:** Introduced a web-based user interface using Streamlit.
- **Key Changes:**
    - Replaced CLI with Streamlit app structure (input fields, buttons, expanders).
    - Added UI feedback (spinners, progress bars) for operations.
    - Incorporated Streamlit caching (@st.cache_resource, @st.cache_data).
    - Default LLM model for Groq updated (e.g., to llama3-8b-8192).

Version 3.0: ragtest_c2.py - Multi-Document Support

- **Core Theme:** Expanded to process XLSX (Excel), CSV, and PPTX (PowerPoint) files in addition to PDFs.
- **Key Changes:**
    - Configuration updated for data_dir and SUPPORTED_EXTENSIONS.
    - Added specific extractor methods for new file types (_extract_xlsx, _extract_csv, _extract_pptx).
    - Generalized processing (process_files) and querying (query_files) logic.
    - Chunks now include file_type and more detailed source_info (sheet/slide numbers).
    - Default LLM model updated (e.g., to meta-llama/llama-4-scout-17b-16e-instruct).

Version 3.1: ragtest_c2-1.py - PPTX Refinement & Global Context Aggregation

- **Core Theme:** Improved PPTX content extraction and changed context aggregation strategy.
- **Key Changes:**
    - PPTX extraction attempts to prepend previous slide's title to current slide content for better context.
    - aggregate_context: Now flattens all retrieved chunks globally, sorts by score, then builds a single combined context string for the LLM.
    - run_query adapted for single unified answer. Streamlit UI conditional rendering improved.

Version 3.2: ragtest_c2-2.py (Internal: ragtest_c2_merged.py) - Advanced PPTX Merging

- **Core Theme:** More sophisticated PPTX slide merging based on content heuristics.
- **Key Changes:**
    - New config pptx_merge_threshold_words to identify "title-like" slides.
    - _extract_pptx: Enhanced logic to conditionally merge sparse title slides with their subsequent content slides, creating slide_text_merged blocks.

Version 3.3: ragtest_c2-3.py (Internal: ragtest_c3_query_adapt.py) - LLM-Powered Query Analysis

- **Core Theme:** Added LLM-based query classification and decomposition for text-based queries.
- **Key Changes:**
    - New helpers: _classify_query ("Simple Retrieval" vs. "Complex/Reasoning") and _decompose_query (breaks complex text queries into sub-queries).
    - run_query: Now classifies the query and, if complex, attempts decomposition before retrieval. Original query is still used for final LLM answer generation.

Version 3.4: ragtest_c2-4.py (Internal: ragtest_c4_dataframe.py) - DataFrame Querying via Pandas

- **Core Theme:** Introduced querying of structured data (CSV/XLSX) by generating and executing Pandas code via LLM.
- **Key Changes:**
    - System now stores loaded CSV/XLSX files as Pandas DataFrames in self.dataframes.
    - _classify_query: Updated to include "Structured Query (DataFrame)" category and confidence.
    - _generate_pandas_code: New helper to prompt LLM for Pandas code generation.
    - run_query: If a query is confidently classified for DataFrames, it attempts to identify the target DF, generate Pandas code, execute it (using exec()), and format the result. Text RAG serves as a fallback.

Version 3.5: ragtest_c2-5.py - DataFrame Specialization & LLM Result Framing

- **Core Theme:** CSV/XLSX files are now handled *exclusively* by the DataFrame path (no text indexing for them). DataFrame results are framed by an LLM.
- **Key Changes:**
    - Text extraction and FAISS indexing removed for CSV/XLSX files.
    - New _frame_dataframe_answer method: Uses LLM to convert raw Pandas output into a natural language answer.
    - Corrected exec() call scope in run_query.

Version 3.6: ragtest_c2-6.py (Internal: ragtest_c2-4_sql.py) - SQLite Backend & DeepSeek Metadata

- **Core Theme:** Replaced Pandas exec() with an SQLite database for structured data; used DeepSeek API for schema metadata.
- **Key Changes:**
  - CSV/XLSX data loaded into SQLite tables via df.to_sql().
  - _get_column_metadata_deepseek: New method using DeepSeek API for column descriptions & SQL types.
  - New LLM-powered helpers: _identify_target_sql_table and _generate_sql_query.
  - _execute_sql_query and _synthesize_answer_from_sql added for SQL pipeline.

Version 3.7: ragtest_c2-7.py - Conversational Context (Chat History)

- **Core Theme:** Integrated chat history awareness for more contextual LLM interactions.
- **Key Changes:**
  - max_chat_history_turns config parameter.
  - _refine_query_with_history: LLM rewrites current query to be standalone based on history.
  - LLM interaction methods (query_llm_with_history, _synthesize_answer_from_sql_with_history) now use ChatPromptTemplate with MessagesPlaceholder for history.
  - Streamlit UI updated to a chat interface.

Version 3.8 & 3.9: ragtest_c2-8.py & ragtest_c2-9.py - PDF Structure Awareness & UI Polish

- **Core Theme:** Enhanced PDF processing with structural tagging; significant Streamlit UI/UX improvements.
- **Key Changes (v3.8):**
  - _extract_pdf: Added heuristics to tag PDF content with potential_section (e.g., "title_page", "table_of_contents") and extract potential_doc_title.
  - Chunks and context headers now include these section tags.
  - LLM prompts updated for "SatSure proposal" context and specific "don't know" responses.
  - Streamlit UI: Major CSS styling for a dark theme; document management moved to a sidebar expander.
  - *Version 3.9 (ragtest_c2-9.py) is identical to Version 3.8.*

Version 4.0: ragtest_c2-10.py - Switched to SambaNova LLM

- **Core Theme:** Replaced Groq (for main LLM) and DeepSeek (for metadata) with SambaNova as the LLM provider.
- **Key Changes:**
  - Configuration updated for SambaNova API (model, URL, key).

- All LLM interactions now use the openai library to call SambaNova API.
- _get_column_metadata_deepseek replaced by _get_column_metadata_sambanova.
- Removed langchain-groq and requests (for DeepSeek) dependencies.