ADVANCED RAG:-

Query Expansion - Take initial user query and rewrite / expand it.
Guess or hypothesize what the answer may look like and provide that in the prompt. This improves our retrieval.

Pitfalls of Simple Vector Search and Spatial Visualisation:-

Embeddings and their vectors are a **geometric data structure**, and we can visualize them spatially. We can project them down into two dimensions.
For this we use **UMAPS - Uniform Manifold Approximation** -> **Projecting high dimension data into 2/3 dimensions.** UMAP tries to maintain the structure of the data.
Project the embeddings one by one -> UMAP is sensible to its inputs. We plot these embeddings to get a visualization of our embeddings.
We are asking it to perform a specific task using only a general representation, which makes things complicated.

Distractors - Retrieved data which doesn’t have a lot of relevance to the query and rather distracts the LLM, giving sub optimal results.
When we pass an irrelevant query we get completely irrelevant results full of Distractors. We need to minimize these results.

QUERY EXPANSION :-

Expansion with Generated Answers:-

Take your query and pass it to an LLM to get an imaginary or hypothetical answer. This answer is then concatenated with the original query, and passed to our vector database for more effective retrieval.
We ask the model to hallucinate and use this hallucination for something useful.

Expansion with Multiple Queries:-

Take your original query, pass it to the LLM and ask it to generate several new related queries. Concatenate all these queries and pass it to the Vector DB for retrieval. For each query, retrieve results from the DB and send unique retrieved documents to the LLM for the final results.

Cross Encoder Re-ranking:-

Score the retrieved results in order of their relevance to our original query.
We pass all the retrieved results to a Reranking model so that the most relevant results have a higher rank. Now we select the top ranking results to be context for our query, and pass it to an LLM.

Cross Encoder model -> Finds similarity score between a query and a retrieved document. This score can be used as a relevancy or ranking result. (BERT Cross Encoder)

Embedding Adaptors:-

Using user feedback about the relevancy of the retrieved results to improve the performance of the retrieval system.
Since we don't have user feedback with us for the RAG application, we first generate some queries similar to our query using the LLM. Then we retrieve documents for each query and pass each document with the corresponding query to the LLM.
We ask the LLM to give a yes or no answer as to whether the retrieved document is relevant to the query or not?
We want relevant results to point in the same direction as 1, and non relevant result to point to -1.

Then we re rank these results and get the most relevant set of queries.