1. Exercise: Designing a RAG system(QnA for manufacturing products)

- **Background**: The company has asked you to create a system User asks questions related to products and you need to design an automated system which can answer these questions.

- **Type of Data:** Unstructured pdf documents

- **Data**:
    - Product Manuals – PDF documents of the product manuals
    - User Queries – List of expected user queries per product
    - Ground Truth – Expected retrieved context and answers for every query
    - Link to data: Read the complete details and data inside “Rag based Qna” of datasets section

- **Expected Result:** The RAG system should be able to answer the user questions about the products

- **Libraries to use:** Langchain
  https://python.langchain.com/docs/use_cases/question_answering/

  Llamaindex and some other libraries can also be used

- **Text extraction packages :** PyPDFLoader, PyPDFium2, PyMuPDFLoader, UnstructuredPDFLoader, PDFPlumberLoader, Unstructured IO

- **Table extraction packages :** Tabula, Camelot

- Databases like BM25, TFIDF, Embedding Retriever, Chroma DB, Milvus, Pinecone, FAISS can be used

- **Open source models**: Dolly, Falcon, Llama, T5, BART, Flan-T5 , Pegasus can be used.

2. Deliverables

- Do the assignment of google colab
- Colab notebook containing the codes
- Provide a report (txt or or doc or docx) file containing following details:

    - Section 1: Metadata about document chunks (i.e. chunking methods used, splitters available with details of splitting techniques, chunk size measurement criteria and chunk size used, whether any kind chunk overlapping used (if used then provide its details), Whether you used any kind of chunk processing or preprocessing for the input text (if any please provide details), etc)

    - Section 2: DB details(only if you used a DB) (i.e. report on database loading time, query embedding time, retrieval time, summarization time, model load time, model GPU/CPU usage. Include any other processing latencies across queries)

    - Section 3: LLM or model evaluation parameters (i.e. tokens per chunk for LLM used, decoding strategy used (in case its a generative model), etc)

    - Section 4: Experiment Results on the best parameters (i.e. What prompt was used (if any), chain types used across queries, best chunk parameters, retriever parameters(if any), five example results, Retrieval level metrics and final answer level metrics (by this time you should know the retrieval level and answer level metrics, refer to the metrics section).

    - Create the docs pointwise

- The colab notebook and the report should be uploaded in the folder named “Assignment 3” on github