Task Overview Develop a prototype for an advanced document analysis system using transformer-based models, incorporating real-time annotation capabilities. This task is designed to be completed in 8-10 hours and should showcase your team's ability to work with state-of-the-art NLP models, handle data processing, and create an interactive user interface.

Project Goals

1. Implement a document ingestion and preprocessing pipeline
    - Take pdf and store in mongo db
    - Upload newly annotated documents to mongo db upon finishing annotation, for the sake of version history and referencing
    - Create one pipeline function which will accept all the text in the initial file and then repeatedly accept new text as real time annotation happens
        - Create a function that accepts a file path and set up conditionals to check what the file extension is.
            - Use PyPDF2's PdfReader module to iterate through pdf documents and use the extract text built in function
            - Use docx module to handle situations where the document is a docx or doc file
            - Can handle text files without external modules in in python
            - Remove metadata and filter using PyPDF 2
            - Have clean text ready for distil bert model
        - Word2vec for vectorization
        - Tfidf for text splitting
        - Use nomic for embeddings
        - Add to Pinecone Vector DB
2. Utilize transformer models for advanced text analysis and embedding
    - Use Distil-BERT, fine tune two separate models, one for NER and one for relation extraction
3. Develop a sophisticated named entity recognition and relation extraction system
    - Call NER pipeline to extract unique entities, and use relation extraction pipeline to analyze relation between every unique pair of entities that were gathered.
4. Create an interactive web interface with real-time annotation capabilities
    - Four primary components:
        - Document Upload, once uploaded you can toggle between extracted text and the actual document
        - Google docs type annotation section
        - Search box to utilize rag in order to bring up information within the document, retrieved info section
        - Generated AI info section, result of NER and relation extraction pipeline
5. Implement a basic versioning system for annotations
    - Mongo DB version control