**REPORT ON DATA ENGINEERING ASSIGNMENT**

1. **Medical Data Management 101**

**Abstract:** Created a python notebook where I wrote some functions to extract data from the dicom files provided in the problem document. Segregated the image from the dicom, saved it in separate output folder in ‘.jpg’ format and simultaneously extracted the metadata for all the respective files and stored in ‘.json’ format as instructed.

**Introduction:** Task is to extract data (image and metadata) from dicom images provided in the link and store them into respective folders.

For the task, I’ve created two separate functions-

1. *convert\_dicom\_to\_jpg()* - To extract image from the dicom file and store it in .jpg file format. All extracted jpgs are stored in a dedicated folder with the same name as of the original file.

E.g., file name: *image-77609639-92081231-39184704-933.dicom*

Output name: *image-77609639-92081231-39184704-933.dicom.jpg*

1. *extract\_metadata()* – To extract metadata from the dicom file and store it in .json file format. All extracted json files are stored in a dedicated folder with the same name as of the original file.

E.g., file name: *image-77609639-92081231-39184704-933.dicom*

Output name: *image-77609639-92081231-39184704-933.dicom.json*

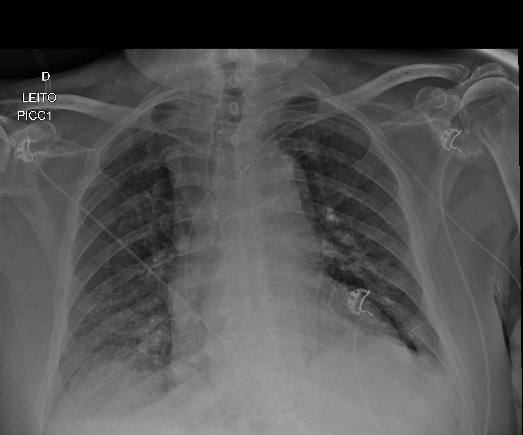
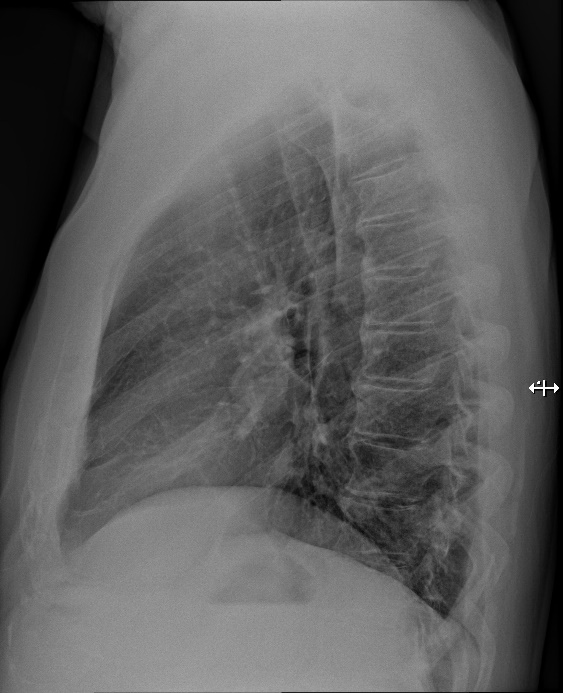
**Project Objective:** The goal of the project is to use the different libraries of python and apply data engineering to create a model to extract data from any kind of dicom files which are present or can arrive in near future having similar file definitions and characteristics.

**My Approach:** Firstly, downloaded the datasets which are present on the cloud and imported it into my Google Colab notebook. In there, created two separate functions to deal with two different scenarios.

* 1st function to import dicom files and extract pixels from the images converted the pixels into float values then with the help of Python’s numpy library rescaled the image and extracting only pixel information from the original file inside the function and converting that information into .jpg files.

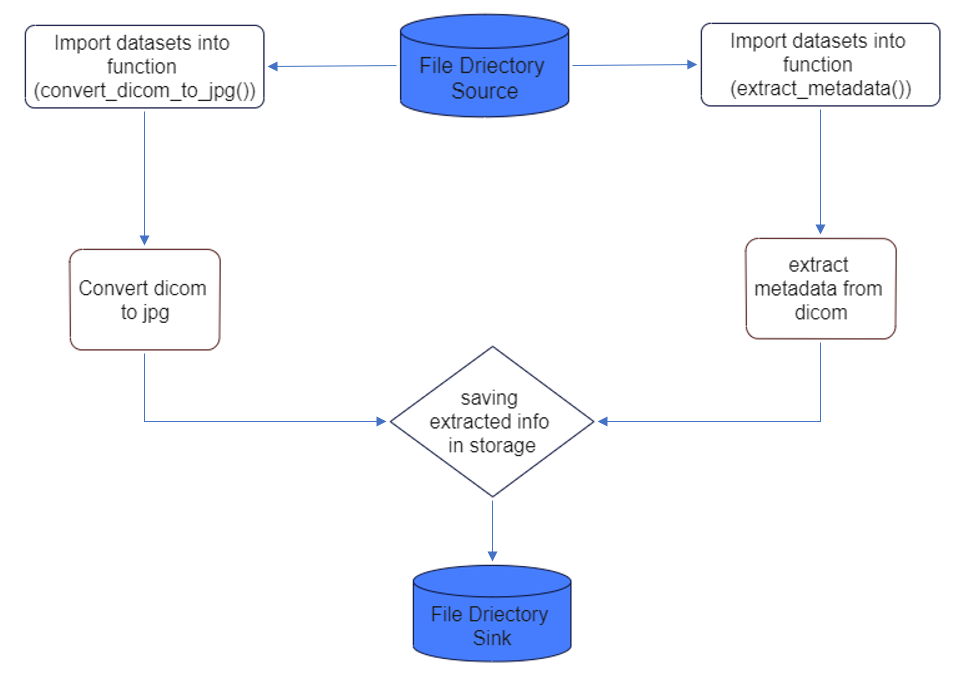
For this conversion I’ve used Python’s **pydicom** library to handle dicom files and **numpy** library to rescale the pixel information into image.

SAMPLE IMAGES FROM THE FINAL RESULT:

* 2nd function to import dicom files and use **pydicom** library’s filereader method to read metadata information and then used another function to load that information in the form of dictionary. After that simply used python **pandas** library to convert that information into a dataframe and saved that dataframe into json files respectively for each and every dicom files individually.

**Architecture:**

****

Link to the notebook:

[qure\_ai\_problems/qure\_ai\_problem\_1.ipynb at main · AbhiRishi96/qure\_ai\_problems (github.com)](https://github.com/AbhiRishi96/qure_ai_problems/blob/main/qure_ai_problem_1.ipynb)

**References:**

[Pydicom |](https://pydicom.github.io/) - To understand and learn about pydicom library.

[NumPy](https://numpy.org/) – Learned about numpy’s feature to rescale pixels into images.

[Extract metadata from DICOM | Kaggle](https://www.kaggle.com/code/sarmat/extract-metadata-from-dicom/notebook) – Got information about a function so that I could store metadata in dictionary format for further processing.

Also took some help from [Stack Overflow](https://stackoverflow.com/) and [YouTube](https://www.youtube.com/) to tackle some difficulties while developing the notebook.

Finally, lots of Googling!!!

1. **Load, Clean and Transform multiple datasets**

**Abstract:** Developed a python notebook where all the datasets provided are loaded into separate dataframes and then I applied several stages of data cleaning methods to produce the result as mentioned in the problem statement.

**Introduction:** Task is to import multiple datasets into respective dataframes to each of the files for further transformation. For this I’ve used Python’s Pandas library with which I cleaned each dataframes, removed noise from the data, made transformations according to the problem statement and merged them into a single unified dataframe with only necessary rows and columns.

**Problem Statement:** The goal of the project is to use the different libraries of python and apply data engineering methods to create a notebook to clean and extract data from files of multiple different formats which are present or can arrive in near future having similar file definitions and characteristics.

**My Approach:** Firstly, downloaded all the datasets from the links provided in the document and stored them into a particular location then provided paths for each of the files to the respective dataframes in the notebook.

I had to deal with three different datasets for which I created dataframes-

* *df\_energy* – for energy indicator file
* *df\_sciem* – for Sciamgo Journal and country rank data file
* *df\_gdp* – for GDP data file from world bank

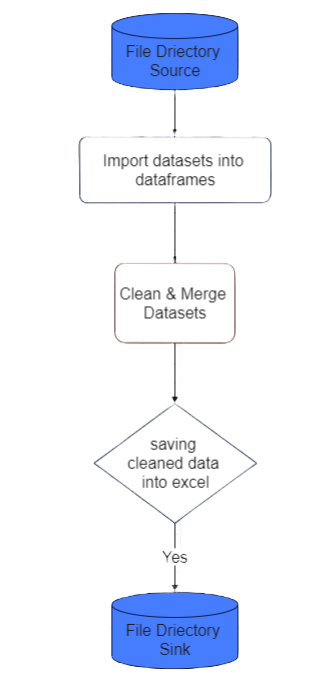
Exactly as stated in the problem.

After this performed data cleaning operation for each of the dataframes.

1. Used pandas .iloc() to select only required data from the *df\_energy* datframe.
2. Renamed the columns as mentioned in the problem statement.
3. Cleaned data, removed special characters from the columns where it was not necessary.
4. Renamed some column names according to the instructions.

After performing data cleaning, finally merged the dataframes with taking ‘country’ column as a common parameter in all the three dataframes, picked top 15 countries from the list according to their rank and finally from *df\_gdp* selected records of last 10 years only and created the final dataframe ‘*df\_final*’.

**Architecture:**



Link to the notebook:

[qure\_ai\_problems/qure\_ai\_problem\_2.ipynb at main · AbhiRishi96/qure\_ai\_problems (github.com)](https://github.com/AbhiRishi96/qure_ai_problems/blob/main/qure_ai_problem_2.ipynb)

**References:**

[pandas - Python Data Analysis Library (pydata.org)](https://pandas.pydata.org/)

[Stack Overflow](https://stackoverflow.com/) for some guidance with approach

1. **Extract data from the image which is inside the box into the excel sheet**

**Abstract:** Developed a python notebook and defined some functions to first extract image file from the pdf file and then another function to extract texts from the image. After that applied some data cleaning to just keep the required information and discarding other noisy data as stated in the problem statement.

**Introduction:** Task is to extract texts from the box inside image which has been attached in the pdf itself. For this, created a notebook and utilised python’s libraries like **PyMuPdf** and **easyocr** to first extract images from the pdf file and then to extract text from the images. Then used **pandas** library to load extracted text into a dataframe, clean the data and store it in excel file.

**Problem Statement:** The goal of the project is to use the different libraries of python and apply data engineering methods to create a notebook to extract texts present inside an image attached to the pdf file.

**My Approach:** Firstly, created a Python notebook in Google Colab, installed & imported required libraries and methods. Loaded the pdf file in which the image is present.

1. Extract the image from pdf:

Used *fitz* module from PyMuPDF library to open the pdf file, then called *get\_images()* to identify all the images present inside the pdf file.

After this, used *extract\_image()* to extract the image bytes from the pages where the images have been identified then loaded the identified bytes into PIL (Python Imaging Library) by using *BytesIo()* and finally saved the image into the local disk / directory by using *save().*

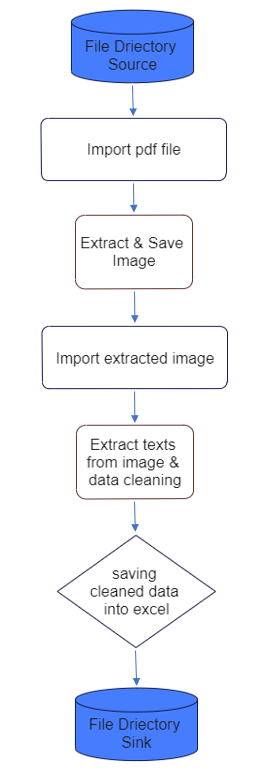
1. Extract text from image and Data cleaning:

After extracting & saving the image in the disk, imported that image into the notebook, then used easyocr’s *reader.readtext()* to extract all text from the image and then load the extracted text into pandas dataframe by using *pd.DataFrame().*

Performed data cleaning operations to eliminate noisy data from the dataframe and keep the data from only the boxed part of the image. For this, used pandas *iloc()* to select data for the specific section, used *replace()* to rename misinterpreted data and provide correct names, used *.loc()* to insert lost data in the dataframe at extact place.

Ultimately used *df.to\_excel()* to save the dataframe into excel file.

**Architecture:**

****

Link to the notebook:

[qure\_ai\_problems/qure\_ai\_problem\_3.ipynb at main · AbhiRishi96/qure\_ai\_problems (github.com)](https://github.com/AbhiRishi96/qure_ai_problems/blob/main/qure_ai_problem_3.ipynb)

**References:**

[PyMuPDF · PyPI](https://pypi.org/project/PyMuPDF/) To read & learn how to utilize the library to extract images from the pdf files.

[easyocr · PyPI](https://pypi.org/project/easyocr/) To understand about the working of the library and what methods to use to read text from an image.

[Extract images from PDF using python PyPDF2 - Stack Overflow](https://stackoverflow.com/questions/20327681/extract-images-from-pdf-using-python-pypdf2/70105425#70105425) took reference from this problem to fix an error which I was getting because of using older methods to get images from a page.

Lots of googling!!!

**LINK TO THE GITHUB REPOSITORY WHERE ALL THE NOTEBOOKS, DATASETS & OUTPUTS ARE STORED:**

<https://github.com/AbhiRishi96/qure_ai_problems.git>