**Programming Assignment**

**Instructions :**

* First create a file with a name “create\_dataset.py” which should take a dataset from Kaggle and create the variable which can print coulumns and headers.

**Example code :**

import kaggle

import pandas as pd

import zipfile

kaggle.api.authenticate()

dataset = "harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows"

kaggle.api.dataset\_download\_files(dataset)

with zipfile.ZipFile(

"imdb-dataset-of-top-1000-movies-and-tv-shows.zip", "r"

) as zip\_ref:

zip\_ref.extractall(".")

movies = pd.read\_csv("imdb\_top\_1000.csv")

print(movies.columns)

print(movies[["Series\_Title", "Overview"]].head(10))

* Next create a python file with a name “clean\_data.py” which should clean the data according to below requirements.

**Code should include below instructions :**

# Which columns will you use?

# Clean your columns

# Concatenate the columns needed for your embedding

# Create new column with concatenated and clean text

* Next create a python file with a name “embed\_and\_store\_data.py” which should take each record from the cleaned dataset file and store the embedded data in the vector database.

**Code should include below instructions :**

"""

- Prepare the text to embed for each reccord of your dataset.

- Create the reccord.

- Clean the text.

- Concatenate fields.

- Choose a Sentence Embedding Model.

- Embed the text generated in the previous step for each reccord.

- Store the embeddings in a vector database (i.e. elasticsearch).

"""

* Next create a python file with a name “search.py” which create a streamlit app which takes input and displays output according to below instructions.

**Code should include below instructions:**

"""

Create an Streamlit app that does the following:

- Reads an input from the user

- Embeds the input

- Search the vector DB for the entries closest to the user input

- Outputs/displays the closest entries found

"""

* Finally these files should run one by one and it should create a semantic search engine from streamlitt app which retrieves data what we search for.