**Project 3**

## Imports

from itertools import chain

import string

from collections import Counter

import pandas as pd

import requests

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

import json

import requests

import json

**Questions 1**

##Make Request

def get\_aspirin\_info(record\_number):

base\_url = "https://pubchem.ncbi.nlm.nih.gov/rest/pug\_view/data/compound"

compound\_url = f"{base\_url}/{record\_number}/json"

response = requests.get(compound\_url)

if response.status\_code == 200:

return response.json()

else:

return {"error": f"Failed to retrieve information for record number {record\_number}"}

# Example usage:

aspirin\_info = get\_aspirin\_info(2244)

print(json.dumps(aspirin\_info, indent=2))

Graphical user interface, text, application

Description automatically generated

**Question 2**

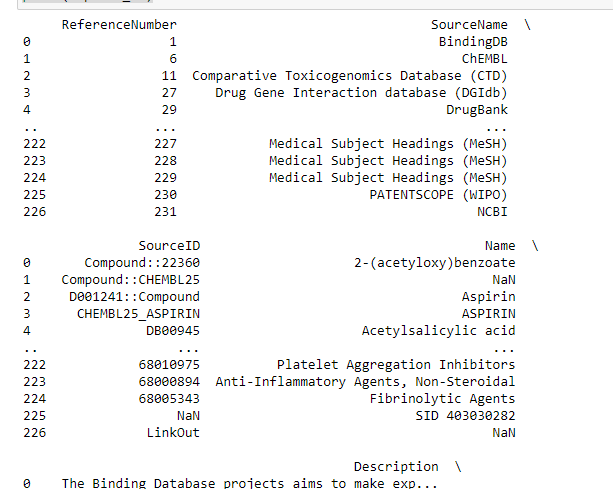
## Set list and create dataframe

aspirin\_refs: list[dict] = aspirin\_info['Record'] ['Reference']

aspirin\_df = pd.DataFrame(aspirin\_refs)

print(aspirin\_df)

aspirin\_df.head()



**Question 3**

##Prepping text and tokenizing

text: str = aspirin\_df ["Description"].str.cat(sep= " ")

tokens: list[str] = word\_tokenize(text)

tokens\_lc: list[str] = [token.lower() for token in tokens if token not in string.punctuation]

wnl = WordNetLemmatizer()

tokens\_lc\_nostop\_lemmatized: list[str] = [t for t in tokens\_lc if t not in stopwords.words('english')]

bag\_of\_words: Counter = Counter(tokens\_lc\_nostop\_lemmatized)

**Question 4**

bag\_of\_words.most\_common(10)

Text

Description automatically generated