**#Python Script for Data Collection**

%pip install requests-oauthlib

import json

import os

import requests

from glob import glob

import shutil

from google.colab import files

import re

import pandas as pd

bearer\_token = "####################################################"

search\_url = <https://api.twitter.com/2/tweets/search/recent>

def bearer\_oauth(r):

r.headers["Authorization"] = f"Bearer {bearer\_token}"

r.headers["User-Agent"] = "v2RecentSearchPython"

return r

def connect\_to\_endpoint(url, params):

response = requests.get(url, auth=bearer\_oauth, params=params)

print(response.status\_code)

if response.status\_code != 200:

raise Exception(response.status\_code, response.text)

return response.json()

def save(data, file\_path):

with open(file\_path, 'w') as file:

json.dump(data, file, indent=4)

def main():

max\_results = 100

tweets = []

next\_token = None

while len(tweets) < max\_results:

query\_params = {

'query': '### -is:retweet',

'tweet.fields': 'author\_id',

'max\_results': 100

}

if next\_token:

query\_params['next\_token'] = next\_token

json\_response = connect\_to\_endpoint(search\_url, query\_params)

tweets.extend(json\_response.get('data', []))

if 'meta' in json\_response and 'next\_token' in json\_response['meta']:

next\_token = json\_response['meta']['next\_token']

else:

break

print(f"Obtained {len(tweets)} tweets")

save({"data": tweets}, 'twitter\_data.json')

print("Data saved to twitter\_data.json")

if \_\_name\_\_ == "\_\_main\_\_":

main()

#loading the datasets

directory\_path = "/content/twitter\_data\_directory"

os.makedirs(directory\_path, exist\_ok = True)

if os.path.exists(directory\_path):

print("Directory created.")

else:

print("Directory could not be created.")

#files to be stored saved into a list

files = ["/content/twitter\_data.json"]

twitter\_data = []

for file\_paths in files:

with open(file\_paths, "r") as file:

data = json.load(file)

twitter\_data.append(data)

#convert json files into excel

merged\_data\_file = "twitter\_merged\_data.json"

with open(merged\_data\_file, "w") as file:

json.dump(twitter\_data, file, indent = 4)

print("JSON files have been merged")

#load data

with open("twitter\_merged\_data.json") as file:

uncleaned\_data = json.load(file)

print("file loaded")

json\_data = uncleaned\_data

df\_columns = ["author\_id", "text", "id", "edit\_history\_tweet\_ids"]

extract\_data = []

for inner\_dict in json\_data[0]["data"]:

extract\_data.append([inner\_dict[column] for column in df\_columns])

print(extract\_data)

#data for EDA

df = pd.DataFrame(extract\_data, columns = df\_columns)

excel\_file\_path = "/content/data\_to\_sort.xlsx"

df.to\_excel(excel\_file\_path, index = False)