Welcome to the assessment test for Lilly E&C Data Scientist.

We at Lilly, deal with lot of structured and unstructured text data. Hence, understanding and analysing textual data, and inferring context is an important aspect of natural language processing (NLP). Your role at Lilly will require you to have a strong working knowledge on text pre-processing, NLP algorithms and applied statistics.

In the next page, we have **four exercises** with an objective to evaluate different aspects of NLP, especially to test your competency to build and evaluate a multi-class text classifier. The points distribution for each of the exercises are provided below:

Exercise 1	10 points
Exercise 2	10 points
Exercise 3	30 points
Exercise 4	50 points
Total	**100** points

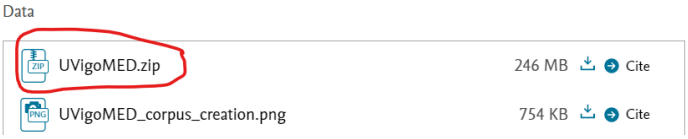
We have provided **Tip** on the way to help you solve this test efficiently.

Expected output(s) is a well commented Jupyter notebook file(s) for model training and inference. At the end of the week, please email your responses and outputs back to jaiswal_geetika@network.lilly.com **.**

Dataset

Please follow the following instructions to access the publicly available dataset:

1. Goto https://data.mendeley.com/datasets/p3jkppwr29/1
2. Download the UVigoMED.zip (246MB) file



3. Unzip the UVigoMED.zip either in Google Colab or on your local laptop/desktop
4. The working folder for all the exercises is **single_label**

5. The **single_label** folder contains two separate subfolders - **train** and **test**, each containing `43979` and `10874` JSON files, respectively. The dataset contains information about abstracts from medical journals and the associated disease categories

Tip Strongly encourage the candidate to use Google Colab for downloading and unzipping the data. This will save you loads of time

Good Luck!!!

Exercise 1

a. Create a table/dataframe which consolidates all the JSONs in **single_label/train** subfolder. Name it as **df_train**. Select only the columns **abstract** and **categories** from df_train
b. Create a table/dataframe which consolidates all the JSONs in **single_label/test** subfolder. Name it as **df_test**. Select only the columns **abstract** and **categories** from df_test

Tip If you are successful in reading the JSONs and converting into tables/dataframes, a sample output is shown below for reference.

abstract	categories
To compare the health care use of workers with...	Wounds and Injuries
Merkel cell carcinoma (MCC) is a rare and aggressive…	Virus Diseases

Note If you are unable to download the data and do needed transformation (as mentioned in Exercise 1 ONLY), please email jaiswal_geetika@network.lilly.com . We will provide you the code snippet to create a tabular dataset. However, exercising this option will incur a **penalty of 20 points.**

Exercise 2

a. What are the top 3 insights generated while doing the data analysis on train set (df_train)?
b. What are the top 3 data challenges you observe on train set?

Exercise 3

a. Pick your favourite machine learning algorithm to train a multi-class text classifier using the train set (df_train). The classifier should be able to consider the abstract as input and predict any one of the 26 disease categories
b. Report key metrics on your test set (df_test) and explain your observations
c. Explain the rationale behind choosing the algorithm in 3.a

---

Exercise 4

Use any one of the Deep Learning frameworks (keras, pytorch, etc.) to:

a. Build a text classifier which classifies the abstracts into one of the 26 disease categories **using any RNN** based architecture and report key metrics on test set. Explain your observations
b. Build a text classifier which classifies the abstracts into one of the 26 disease categories **using any Transformer** architecture and report key metrics on test set. Explain your observations
c. You are free to experiment various LSTM and Transformer architectures for 4.a and 4.b, however, only report the model which you consider the best. What is your rationale for this model selection?
d. Consider the test table/dataframe (df_test). Run an inference through the best model determined in 4.c. What are words/phrases from the abstracts that drive the predicted category?

Tip The assessment team was able to build the text classifier and infer on Google Colab

---