**Statement of Purpose**

The purpose of this document is to go over the high-level solution design for the clinical data exercise.

**Problem Statement**

Apply common Machine Learning techniques on relevant sections (chief complaint, history of present illness, and discharge diagnosis) clinical data (clinical\_notes\training\_data\\*.txt) to uncover common underlying factors for a given medical condition.

Execution steps

1. Unzip the **boris\_kundu\_assignment.zip** file to some location on your local system.
2. Install Anaconda on your system and launch Jupyter Notebook.
3. From Jupyter console, navigate to the unzipped directory (from step 1) and select the **Topic\_Modeling\_Boris\_Kundu.ipynb** notebook to open it.
4. Uncomment the pip commands shown below or install manually.

**#!pip install --upgrade pip**

**#!pip install pandas**

**#!pip install numpy**

**#!pip install nltk**

**#!pip install spacy**

**#!pip install sklearn**

**#!pip install pyLDAvis**

**#nltk.download('wordnet')**

1. Please note that n\_jobs is set to 4 by default for parallel execution during hyperparameter tuning. Kindly, change it if needed under **Modeling** section.

(best\_lda\_model, gs\_model) = hyperparameter\_tuning(pipeline\_LDA, params\_LDA, df['pos\_filtered\_text'], jobs = 4)

1. Add/update inputs in last caell of **Prediction** section (last cell) as per your choice to get topic words for them.

**list\_of\_conditions = ['degenerative joint disease', 'loss gag reflex', 'bright red blood']**

1. Please note the custom stop words have a significant impact on the topics being assigned. Kindly, add/update same as per need in first cell of **Data pre-processing** section.

# Corpus specific stop words

custom\_stop\_words = ['history','date','birth','sex']

1. After making changes as per need. Click on Cell -> Run All to execute the code.

**High level solution design**

Please refer to the HTML/PDF version of the notebook in **boris\_kundu\_assignment.zip** to see the description, comments, and output for each cell.

1. *Setup libraries for implementation*
   1. Install packages
   2. Import packages
   3. Download other libraries (as needed)

The main packages used here were:

NLTK: Used for tokenization, stop words, lemmatization, and POS tagging.

Scikit-Learn: Used for vectorization, topic modeling, and hyper parameter tuning.

1. *Load data from files*
   1. Define relevant sections
   2. Parse given text files
   3. Create raw dataset

The relevant sections parsed (as mentioned in the problem statement provide) from all the .txt files were chief complaint, history of present illness, and discharge diagnosis. A raw pandas data frame was created from this data.

1. *Data pre-processing*
   1. Clean data (punctuations, white spaces etc.)
   2. Tokenize text
   3. Remove stop words
   4. Lemmatize text
   5. Filter out unwanted Parts of Speech

Data was cleaned to remove non-alphabetical and new-line characters. Trailing and sequence of white spaces were also removed. After this, tokenization was done and stop words including some custom ones were removed from it. Words were reduced to their base form using lemmatization and only POS tags matching noun, verbs, adverbs, and adjectives were kept.

1. *Modeling*
   1. Setup pipelines for model
   2. Perform hyperparameter tuning using pipeline
   3. Use best parameters for models to perform topic modeling

Pipeline was initialized with vectorization and topic modeling models. Hyperparameters such as max\_df, ngram\_range, n\_components etc. were defined for tuning using GridSearchCV. The best parameters obtained were used for topic modeling. Additionally, topics were visualized to understand the features/words in them.

1. *Predictions*
   1. Identify topics (underlying factors) for existing documents (conditions)
   2. Identify top features (factors) in each topic
   3. Predict topic (underlying factors) for a new text (condition)

Main topics were identified for each document along with document count per topic. Top words per topic were also identified. Finally, topic predictions were made for new custom inputs and relevant words/features are displayed.