SmartEHR

Background: Electronic Health Records or EHR is a digital version of the patient's paper chart. EHRs are real-time, patient-centered records that make information available instantly and securely to authorized users. While an EHR does contain the medical and treatment histories of patients, an EHR system is built to go beyond standard clinical data collected in a provider's office and can be inclusive of a broader view of a patient's care.

Problem statement: In this project I attempted to leverage the power of text analysis to extract patient-specific information from EHRs. The **dataset** used for this purpose are the **discharge summaries** in the form of free-text clinical notes. Each of these notes have been compiled from the notes written by various sources like nurses, doctors, house-staff and other medical personnel that were assigned to a patient during his/her stay. As a result, for a particular patient there can be **multiple diagnoses entries.**

The algorithm built as a part of this project tries to achieve the following three tasks using **only** clinical notes as training data:

1. Predict the chances of a patient getting re-admitted in the hospital within 30-days of their discharge.
2. Predict the diagnoses codes assigned to the patients during their stay in the hospital
3. Extract important keywords that are very specific to a particular patient

The inspiration for this project is mainly three-fold. In many countries, hospitals are penalized if a patient is re-admitted within 30 days of discharge. So, it's an important task to predict the chances of re-admission. Secondly, just by looking at the clinical notes, can a machine learning model be trained accurately to predict the diagnoses and keywords? This can then be used by medical

personnel as an efficient text summarizer that summarizes huge discharge notes into a set of keywords.

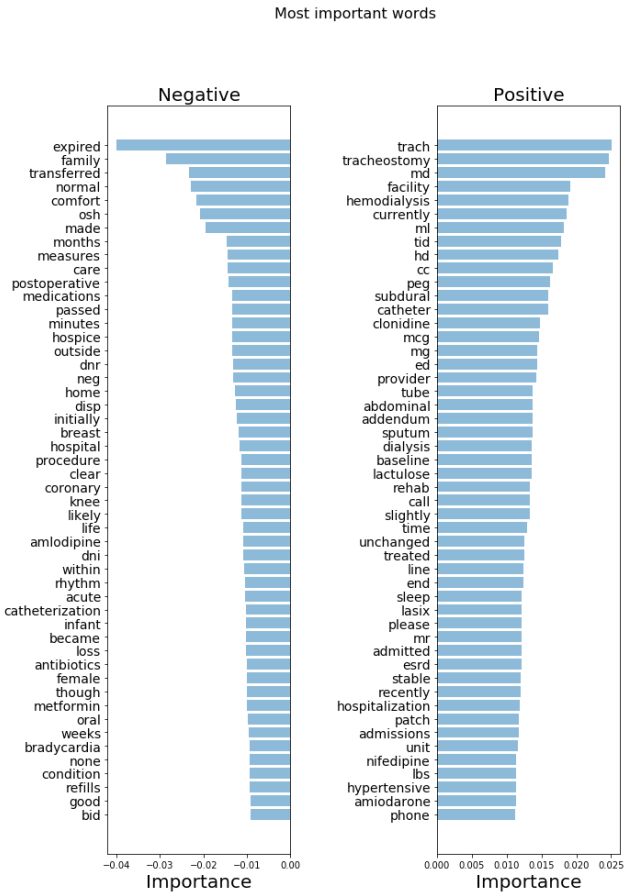
Dataset: We considered MIMIC III (Medical Information Mart for Intensive Care III) free hospital database. This database contains de-identified data from over 50,000 patients who were admitted to Beth Israel Deaconess Medical Center in Boston, Massachusetts from 2001 to 2012. In order to get access to the data for this project, you will need to request access at this link (https://mimic.physionet.org/gettingstarted/access/)

Model: We considered only **"Discharge summaries"** as our primary source of data. Each patient's summary text was extracted and mapped with the corresponding admission and discharge dates in that hospital. Correspondingly, we added a new column to our dataset called **"DAYS_NEXT_ADMIT"** that contained the number of days between admission. A binary class label (>30days: 1 and <30days: 0) was assigned to each datapoint. Also, the corresponding diagnoses codes (ICD9) were collected in a separate column called **"ICD9_CODE"**.
So, our final dataset contained six main columns:

1. Patient id: SUBJECT_ID
2. Hospital admission id: HADM_ID (to track the number of hospital visits)
3. ICD9_CODE: diagnoses codes
4. Binary class label: OUTPUT_LABEL
5. Free text discharge summary: TEXT

Text cleaning and model building: Standard pre-processing tasks were performed before building the model. A CountVectorizer and a TF-IDF vectorizer was used to represent the text in the form of vectors for further processing. The model was saved for future use and a logistic regression classifier was trained on the text and the

corresponding labels. Standard “stopwords” from the **nltk** package was used to not include certain commonly used words in the model. The model was judged for its accuracy using the standard 33-66 train/test split.



30-day readmission: Certain keywords that the model used to classify between the two classes (Positive: <30-day Negative: >30-day or never) was displayed.

From the above figure, some of the most informative words used to describe positive class are **esrd, peg, subdural, tracheostomy etc.** These were in turn tested for literature reference and the following links showed that they are indeed associated with an increased 30-day readmission rate.

Esrd: https://www.medpagetoday.com/nephrology/esrd/68196

Peg: https://www.ncbi.nlm.nih.gov/pubmed/27423366

Subdural: https://www.ncbi.nlm.nih.gov/pubmed/29271714

This is just an indication that our model is indeed able to capture important words that are associated with the positive label: **<30days**

Diagnoses codes: The top-30 most occurring diagnoses were considered. Since there are multiple codes/diagnosis associated with each patient, a multi-label classification approach was followed, where each logit model was trained on each class and saved for future use. Given a discharge text, each of these models will then output the probability of a particular disease occurring. The top-10 most probable diseases for each patient is then printed on the screen.

Keyword: Given a huge discharge summary, can we extract meaningful keywords from the text that is very specific to the text and not to the entire training corpus? This was the main idea behind keyword extraction. A TF-IDF vectorizer was ideal for this case and it efficiently takes into account low frequency words specific to a document and not consider high frequency words like “and”, “the”, “patient”, “admission” etc.

Tech-stack used: A python-flask application was developed to deploy this as a web-app. All the model files were saved in .pkl format for later usage. The **README.txt** file lists all the dependencies and instructions required to run the app. **Four test discharge summaries** are also included in the submission folder. Some parts of the web app were made using Bootstrap library and the diagnoses codes have been hyperlinked with the corresponding ICD9 disease web page for ease-of-access. The app takes seconds to compile and print the output on the screen. It is easily deployable to a standard web-server and can handle discharge summary text of any length.

A demo video is also included for demonstration purposes. In case of any difficulty running the scripts, please contact me using the email address (bt16s001@smail.iitm.ac.in).

Two ipython notebooks were also included which contained the model building codes