Predicting Hospital Readmission with Discharge Summaries

By

Jerad Williams

Problem Statement

Optimizing hospital readmission lets the hospital manage unattended admissions effectively and have better resource utilization. Prediction of hospital readmission at admission time, within 24 hours, lacks information that would be included in discharge summaries. Discharge summaries are free notes given by doctors which include the crisp description of the disease, and the final discharge notes.

The dataset used for the project is MIMIC-III (Medical Information Mart for Intensive Care III) database, which contains information on about 50,000 patients admitted to Beth Israel Deaconess Medical Center in Boston, Massachusetts from 2001 to 2012. The sensitive information like patient’s name and doctor’s name is replaced by NamePattern keyword, and hence, only the pattern specific to disease and pattern obtained from discharge summary are included while considering building a model for the hospital readmission.

The dataset is from a real world scenario which could essentially be used in hospitals to define the budget and make better decisions for the current patients arriving, such as: taking precautionary measure beforehand, while taking care of the patient, or warning the patient beforehand while discharging them, so that future event could be handled with care, which basically not only improves the health of the patient otherwise, but also improves the hospital resource utilization and resource allocation for the actual patient.

The problem statement is entirely based on how to improve hospital resource utilization and reduce cost of unplanned readmission solely on discharge summaries. NLP is a tricky field, and the decision taken in Natural Language for humans is based on vast knowledge associated with the current query. Discharge summaries can essentially help get an overview of the patient’s important details (if not all) to create a model that implies considerate causation. Modern Deep Learning methods help to create strong representation from words, sentences, paragraphs, etc into a vectorized form. This feature representation helps to create classifiers on top of it, which are needed here.

The model will give a probability score for the patient being admitted once again within 30 days to the same hospital based on the discharge summaries. Based on the model’s predictive power, and knowledge it has instilled from the dataset mentioned above (with strong feature engineering), one can predict in terms of percentage whether the patient is going to get admitted or not. This percentage helps to make a better business decision for the hospitals. Based on the aggregated values, hospitals can quantify the various correlation and causation between doctor’s treatment, disease, and discharge summaries.

Data Wrangling Steps

Discharge Summaries are found in the NOTESEVENTS.csv file which contain the following columns: 'ROW_ID', 'SUBJECT_ID', 'HADM_ID', 'CHARTDATE', 'CHARTTIME', 'STORETIME', 'CATEGORY', 'DESCRIPTION', 'CGID', 'ISERROR', 'TEXT'. The only column of interest is ‘CATEGORY’, which provides what type note was written. From here, we will detail if it is a DISCHARGE SUMMARY note, SURGERY note, etc, because there are obviously various notes written for the patient during their stay at the hospital. We filtered CATEGORY == ‘Discharge summary’ for our analysis. The data was pretty clean, except there were multiple whitespaces, so we removed those. I tried to get some structured information from the dataset that was common in almost all of the rows, such as service, and removed sensitive values which were masked in the form of [** NamePattern **] for patients and doctors, which contributed to noise rather than strengthening the information for the discharge summaries. There are multiple summaries for each admission of the patient, however, we are keeping only the last summary, and there are two reasons for this. First, the last summary captures the complete information, which the first or intermediate won’t be able to, and gives the conclusion of the patient's condition. Second, is listed while explaining the ADMISSIONS.csv file structure. Apart from this, after extracting the service from the discharge summary, we clean the service for any extra punctuations and keep only the exact words. The discharge summary is tokenized, punctuations are removed, and words are stemmed and lemmatized, to get a very clean and root form of the words.

The ADMISSIONS.csv file contains the columns: 'ROW_ID', 'SUBJECT_ID', 'HADM_ID', 'ADMITTIME', 'DISCHTIME', 'DEATHTIME', 'ADMISSION_TYPE', 'ADMISSION_LOCATION', 'DISCHARGE_LOCATION', 'INSURANCE', 'LANGUAGE', 'RELIGION', 'MARITAL_STATUS', 'ETHNICITY', 'EDREGTIME', 'EDOUTTIME', 'DIAGNOSIS', 'HOSPITAL_EXPIRE_FLAG', and 'HAS_CHARTEVENTS_DATA'. The SUBJECT_ID indicates the patient ID and HADM_ID indicates the patient admissions ID. The data for this is pretty clean, except the data that had some missing values, which were in the following columns: DEATHTIME, LANGUAGE, RELIGION, MARITAL STATUS, EDREGTIME, EDOUTTIME, and DIAGNOSIS. The target feature (admit within the next 30 days) would be obtained from the ADMITTIME, DISCHTIME for each patient and each admit ID. Hence, we obtain the NEXT_ADMITTIME, and it has missing values as well, because of the fact that some patients are admitted just once only. To fill the missing values with correct values, we employ the data exploration step, and check which values are most likely to occur if the value was missing. So, for DEATHTIME, we are not using that as a feature, simply because there is a flag HOSPITAL_EXPIRE_FLAG which indicates whether a patient expired in the hospital or not. For LANGUAGE, we are choosing the maximum value to fill the missing value. For RELIGION, we are choosing NOT SPECIFIED as default for missing value. For MARITAL STATUS, we are choosing UNKNOWN (DEFAULT). For DIAGNOSIS, we are choosing UNKNOWN. The

date-time features are not directly used, because the date is not correct, but the difference in two dates is preserved for getting correct information. We created features like TARGET (whether the patient gets admitted within next 30 days or not), DAYS_IN_HOSPITAL (how many days a patient stays in hospital), DAYS_WITHIN_NEXT_VISIT (how many days patient takes to get admit for the consecutive time), DAYS_IN_EMERGENCY_WARD (how many days the patient was kept in EMERGENCY Department). The NaN values in DAYS_WITHIN_NEXT_VISIT is filled up with 2*maximum value so that it is not in the dataset. The NaN values in DAYS_IN_EMERGENCY_WARD is filled up with 0.
The ADMITTIME and DISCHTIME are also present in Summary notes which are eventually removed from the notes. Also, for multiple summary notes of the same admit ID, the last DISCHTIME is there in the ADMISSIONS.csv file, and not the intermediate ones. So the last summary is only kept out of multiple summaries for the same admit ID.

Data Story

The admissions file has various columns related to personal characteristics like Ethnicity, Religion, Marital Status, Language, Discharge Location, Admission Location, Diagnosis, and Insurance. The TARGET variable which was created out of the ADMITTIME and NEXTADMITTIME, consisted only 4.45% of the total 58976 Admissions.

The data is highly skewed. The datatypes of the columns were converted to labels and to one hot representation as well, and corresponding correlation were found with the TARGET variables. The hypothesis of the questions made during these were mainly pertaining to the above columns, like whether people with insurance are likely to get admitted again, or the number of days a person was admitted to the hospital also states the chances of them getting admitted again.

Another important question that I thought of was that the discharge location depicts the last treatment the patient undergoes while leaving the hospital. We check the association of all the above columns that are explained in the Exploratory Data Analysis part with the aid of charts such as BoxPlot, Distribution plot, bar graph, correlation Heatmaps.

The time in the emergency ward as well reflects upon the ability to get the patient readmission, and I obtained this feature from the Emergency In and Out time in days because the TARGET value, which is 0/1, is obtained in days. With TARGET it is slightly positively correlated, but comparing it to the DAYS_WITHIN_NEXT_ADMIT gives negative correlation and this is expected as the DAYS_WITHIN_NEXT_ADMIT is more as the time within the emergency ward is less or 0, or if the DAYS_WITHIN_NEXT_ADMIT is less, then the TIME_IN_EMERGENCY_WARD is more, so it is slightly negatively correlated. Also, the correlation between the Insurance and the time in the emergency ward is positively correlated to the “Medicare” Insurance, while negatively correlated to the “Private”.

The correlation between the Diagnosis and the readmission should be meaningful but for rare diagnosis and in some cases where correct treatment is not given or taken, all these conclude to explore the features of diagnosis, discharge location, and days in the hospital with the target variable of days between 2 consecutive admits.

Word Cloud was created to check the most frequent words occurring in Diagnosis. The service was extracted from the Discharge Summary. Most of the words were common in both the cases of TARGET values, except few words which were present more in TARGET == 1 than TARGET == 0.

Apart from this, the CountVectorizer was applied on the Discharge Summary to get the most common words in the summary text and they were related to prescription which is common like a table, mg, day, etc. A feature was made as ‘FOLLOW_UP’. If follow up word is there in discharge summary, we set this to one, and the question we asked pertaining to TARGET, DAYS_WITHIN_NEXT_ADMIT, TIME_IN_EMERGENCY_WARD, and the results as similar to what TIME_IN_EMERGENCY_WARD had with the TARGET and DAYS_WITHIN_NEXT_ADMIT, the correlation of FOLLOW_UP and TIME_IN_EMERGENCY_WARD is slightly positive and more than the TARGET value. This means that doctors advise to Follow Up but the patient does not follow up regularly or the treatment is well given.

Another analysis is on the DIAGNOSIS TF-IDF features to the TARGET values. And the correlation found in this was that as the DIAGNOSIS contained 'FAILURE' in it, the correlation was slightly positive and as it contained 'NEWBORN' in it, the correlation was slightly negative. This makes sense because if there is a DIAGNOSIS of some organ failure, then there is a high chance of getting admitted again.

The same analysis could be done with the WordCloud method and this was done on the Service column obtained from the Discharge summary, which indicated the presence of MED, Neurosurgery to be present in the TARGET == 1, and other disease services like Cardio, etc in TARGET == 0.

The largest length of the summary is around 6000-7000 words according to the distribution plot. I tried to obtain the principal component with the high sum of variance ratio, and with 1000 components the sum of variance ratio gave to be ~.59. The components with a value of 3000-4000 could give better results in terms of variance ratio.

Exploratory Data Analysis

The questions asked in the Data Story part are answered using the correlation values, box-plots, bar graphs, word clouds, taking tf-idf features of words, and doing regression tests.

The DAYS_WITHIN_NEXT_ADMIT feature, which analyzed for its 30 day time-line, showed that maximum re-admissions are on average within 15-20 days, which the BoxPlots of: INSURANCE, ADMISSION_TYPE, ADMISSION_LOCATION, DISCHARGE LOCATION, and LANGUAGE, confirm.

The string columns are converted to one-hot representation, and then a regression analysis was made with the TARGET variable. The simple regression tests between independent variables show that none of the variables explain each other from the fact that the R2 score of the regression model trained is less than 0. Also, classification with each independent feature alone was done which also did not give good results (AUC Score = 0.5), concluding that the features alone does not give much information to classify for the readmission.

Correlation of one-hot representation of Insurance with days in hospital and TARGET values were obtained and it showed that Medicare insurance was highly correlated to both of the above features and Private was least correlated. The correlation of Insurance with Days in the emergency ward is interesting, in that Medicare is more correlated than both the above features and the same with Private as well.

The Boxplot of Admission Type shows that the mean of the Newborn types lies in the 0-9 days of re-admission, which makes sense, the highest mean in Admission Type is of ELECTIVE which means a planned admit to the hospital. The URGENT and EMERGENCY have similar means. The EMERGENCY feature is positively correlated to TARGET values. And the others are slightly negatively correlated. Here, correlations are near zero only. One reason for this could be the fact that the re-admissions comprise just 4% of the total admissions.

Patient demographics do not show any strong correlation with the TARGET value, and that perhaps shows that while readmissions occur, it is more about what disease they have and what treatment they are exposed to, and more on their health conditions.

The word cloud used here, shows a visual representation of the most frequent words used in the service, diagnosis. We found that MEDICINE, CARDIOTHORACIC, SURGERY, and other very sparse words were used for TARGET = 1. For TARGET = 0, the words plotted are MEDICINE, SURGERY, ORTHOPEDIC, which means either the readmission is after 30 days or none.

The word counts graph for discharge summary after removing stop words and cleaning it, shows the bigram of Family History in top 10 bigrams, which means that doctors while doing a treatment for the patient consider family history into account, because most of the services offered are in terms of Surgery, Neuro Surgery, Cardiac surgery, medicine, etc. and in these cases there are chances of re-admissions. So, the text in this sense gives more information about the re-admissions.

In-depth Analysis (Machine Learning)

We are working with features like Tfidf and Word2Vec vectors, tried and tested on models like Logistic Regression, Bernoulli Naive Bayes, Decision Tree, LSTM Based Deep Learning model, and XGBoost, among which XGBoost with complete dataset and subset of features used gave the best result of 0.73 AUROC Score. Following are the details of all models tried.

1. Tfidf Features
    a. We took Tfidf features of Service (obtained from summary text)
    b. Tfidf Feature of Summary text (obtained after cleaning text)
    c. Tfidf Feature of Diagnosis column
2. One Hot Feature Vector Representation

    a. ADMISSION_TYPE
    b. ADMISSION_LOCATION
    c. DISCHARGE_LOCATION
    d. INSURANCE
    e. LANGUAGE
    f. RELIGION
    g. MARITAL_STATUS
    h. ETHNICITY
    i. DESCRIPTION

3. Float Features
    a. TIME_IN_EMERGENCY_WARD (feature created from difference of Emergency Ward In Time and Emergency Ward Out Time in days)
    b. In_failure (feature created from the Diagnosis column, whether failure word is there in Diagnosis column)
    c. FOLLOW_UP (Feature created from the summary text, whether follow up word is there in the summary text)
    d. Sentiment (feature created for sentiment of the summary text)
    e. Subjectivity (feature created for the subjectivity of the summary text)
4. Word2Vec Features
    a. The Word2Vec features are in fact averaged over to get essentially Doc2Vec features. (feature and model created for training Word2Vec model using Fasttext on the summary text)

The models are trained on the above features or subset of the above features. The dataset is split into train and test, with 80-20 split.

The Logistic Regression model is trained on all the above features, with summary max_features of 3000. The AUROC Score obtained is 0.7171 on the test dataset.

The Decision Tree model train on all the above features gives an AUROC score of 0.5168.

The mean of the scores from the above 2 classifiers is obtained and the AUROC score of this is 0.7073. This gets reduced, which means the Logistic Regression prediction scores were around 0.5 values (which means the classifier is not confident).

The Bernoulli Naive Bayes model with all the above same features gives 0.6424 AUROC Score, and the mean of the above classifier predictions with Bernoulli Naive Bayes gives 0.6946 AUROC Score. Technically giving the Baseline Score of 0.7171.

LSTM model is trained on the only cleaned text data with custom Embedding Layer of dimension 300, 128 LSTM cells, and output layer of 1 node with sigmoid activation function. The model was trained with 1 epoch and gave the AUROC score of 0.5720. The LSTM model on the mean of the Word2Vec embeddings trained using FastText Library (essentially Doc2Vec embeddings) trained on an LSTM model of 90 LSTM cells, 64 Dense Layer nodes, and 1 output node with sigmoid activation functions trained for around 300 epochs gave an AUROC score of 0.6721. For the later model the embeddings are scaled in the range of 0 - 1 (MinMaxScaler). The MinMaxScaler is applied so that all the feature values are in the same range and training is more stable and correct with that. Also for the later model the majority class is downsample to 3 times the number of the minority class giving the new distribution of 75%-25% (0-1) as to the old one 96%-4% (0-1).

For downsampling we randomly sample the majority class values so that the downsampled class values are 75% of the total dataset.

The results test data with the training of downsampled data on Naive Bayes was 0.6215 AUROC Score (which is not an improvement given all the features vs just the Doc2Vec embedding features).

The number of features are tried with the XGBoost model as well. The XGBoost model trained with all the above features except the Diagnosis feature vector (Diagnosis is already included in detail in summary text) and with all the given dataset gave the best AUROC score of 0.73. The XGBoost model trained with only Doc2Vec embeddings with the downsampled data gave only 0.6690 AUROC Score. The XGBoost model was tried with different summary text max features for the TfIdf Feature vectors, the maximum we could achieve was with 5000 max features values that is 0.73 AUROC Score, which is just 0.02 less than the State of the Art method which has score of 0.75-0.76.