# Predicting ICU Readmission From Clinical Notes Using NLP and Machine Learning

**Caroline Dickey, Bingyu Fan, Michael Mow, Eliza Tadley**
**Georgia Institute of Technology, Atlanta, GA**

## Abstract

*Machine Learning techniques like Natural Language Processing (NLP) can be used to draw insights from large amounts of unstructured data including medical records. The benefits of analyzing electronic health records (EHR) for patterns are numerous, but the process of extracting meaningful results requires extensive pre-processing and text transformation prior to introducing traditional machine learning models. In this paper, we describe a systematic approach to predicting hospital readmission using a SQL/BigQuery pipeline, feature tokenization and vectorization, and multiple classification models. A model combining lemmatization, a TF-IDF vectorized model, and a tuned logistic regression model produces the best results at predicting ICU readmission within 45 days with an AUC of 0.67.*

## Introduction

The introduction of electronic health record (EHR) systems starting in the 1960s allowed far more healthcare information to be stored than ever before. EHRs contain a wide range of data from medical history, medications, allergies, and laboratory results to demographics and insurance information. Structured data, such as lab results and vitals, is easy to extract from EHR systems. However, EHR systems often contain much more unstructured data such as clinical notes than structured data. The information in these notes can provide more detail and context than any structured data and can be incredibly valuable in diagnosing and treating diseases, improving healthcare outcomes, and reducing cost. However, extracting and processing data from clinical notes is time consuming and expensive. Natural language processing (NLP) and machine learning can be used to create a framework to more quickly and efficiently access unstructured data in EHRs, standardize the data, and use it for analysis.

## Problem Statement

In particular, our aim is to use NLP and machine learning models to predict unplanned readmission to the ICU following a procedure. (We chose to focus on readmission occurring within 45 days of the initial visit, though future work could involve assessing shorter or longer periods of time.) This is particularly important because unexpected readmission is associated with worse health outcomes for the patient and increased costs for the hospital. Predicting readmission before it happens can allow the patient to get additional care and treatment.

## Background and Related Work

Machine learning and natural language processing (NLP) have become hot topics in the healthcare domain over the last several years with new methodologies constantly being developed and tested. To guide our approach, our team performed a survey of research covering existing NLP techniques for healthcare and the issues researchers are facing.

Basyal et al.[1] and Velupillai et al.[2] examine the current state of natural language processing in the healthcare domain including its applications and limitations. In [1] the authors perform a comprehensive survey of the state-of-the-art NLP techniques within the healthcare domain and how to apply them in practice. According to Basyal et al.[1], NLP methods have been successfully used to associate phenotypes, disease symptoms, and drug symptoms, as well as disease classification.

Several state-of-the-art deep learning models that use EHR data for predictive modeling have been built by Rajkomar et al.[3] Classification for in-hospital mortality, unplanned readmission, prolonged length of stay, and discharge diagnoses are presented. Using AUC (Area Under the ROC Curve) as the classification metric, predicting readmission proved to be the most difficult task thus warranting further work in this area.

Previous research has utilized the MIMIC III data set in particular to build NLP models. Huang et al.[4] utilize

bidirectional encoder representations to represent clinical notes in various ML models to help solve the problems of high dimensionality and sparsity in the text. Teo et al.[5] present a framework to quantify the savings from applying machine learning models to predict hospital readmission. Their work expands on the research of others developing readmission models on the MIMIC data set to consider the clinical impact and cost savings of these models.

Previous research into predicting hospital and patient readmission includes a a cost-sensitive long short-term memory (LSTM) network from Ashfaq et al.[6] that utilizes both human and machine-derived features such as lab and diagnostic codes along with a financial analysis to estimate possible cost savings if such a model were implemented in a real clinical workflow. Prediction was done using test patients fed into the trained network to calculate the readmission prediction. The authors evaluated their models using AUC along with F1-scores and cost-savings and found that they could achieve fairly high discrimination ability with an AUC of 0.77 and an F1-score of 0.55[6].

A survey of other models for readmission risk prediction studied by Artetxe et al.[7] found that out of 77 studies, only 19% of the models reported AUC scores above 0.75. Complicated deep neural networks was the best performing model, but it is also the most difficult to manage and interpret. They also reviewed the common types of feature selection techniques and ways to deal with class imbalance among the studies. According to them, recent studies introducing machine learning techniques report promising results and anticipate advantages over classical methods[7].

**Approach and Implementation**

**Data**

The data set for this project is the MIMIC-III (Medical Information Mart for Intensive Care) data set maintained by MIT[8]. The data consists of de-identified health data from over forty thousand patients from the critical care units at Beth Israel Deaconess Medical Center that was collected between 2001 and 2012. MIMIC-III is a relational database that contains information such as demographics, vital signs, laboratory results, procedures, medications, clinician notes, and reports for a wide range of patients. The data is available in both CSV and cloud formats. MIT hosts the MIMIC-III database on both the AWS cloud and the GCP cloud to allow researchers easier and cheaper access to the data. For our project, we are utilizing the MIMIC-III data set via GCP with BigQuery.

The data set consists of 26 tables. A complete list and description of the tables can be found in the MIMIC documentation[8]. For this project, we are primarily concerned with a subset of the tables that include: PATIENTS, NOTEEVENTS, and ADMISSIONS. The NOTEEVENTS table is the most important for our project because it contains a free text field with clinician and caregiver notes in unstructured form. The ADMISSIONS and PATIENTS tables are useful in identifying patients that have been readmitted to the ICU within a specific time frame.

The NOTEEVENTS table contains 2,083,180 records and the main columns of interest are: SUBJECT_ID, HADM_ID, CATEGORY, and TEXT. The TEXT field contains the clinical notes. The notes vary widely in length. A histogram of the length of the notes as seen in the figure below (Figure 1) shows that there are some notes with almost 50,000 characters and some with close to 0 characters. The second histogram below (Figure 2) shows the count of notes by category. Nursing notes are the most common type.

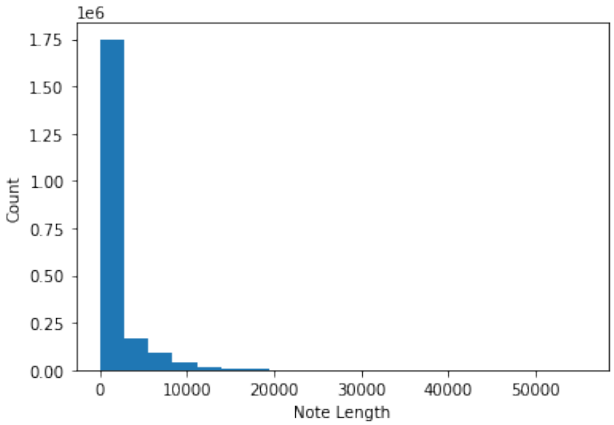


Figure 1: Histogram of the Length of Clinical Notes

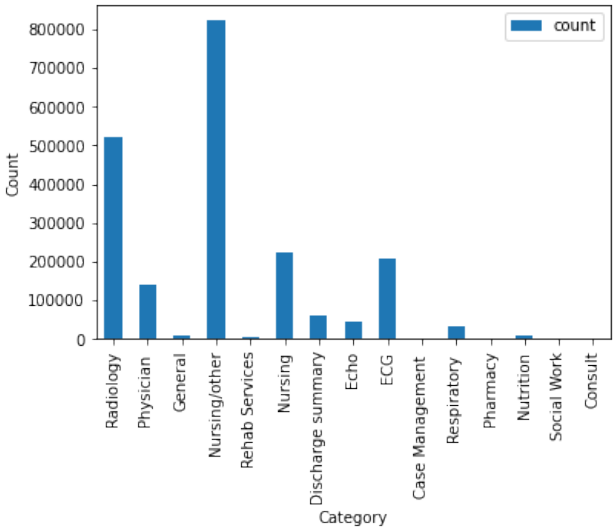


Figure 2: Histogram of the Category of Clinical Notes

To get a sense of the type of information contained in the clinical notes below is a randomly sampled note.

“pneumonia, resp failure micu npn o- afebrile po. cv- hr 70-90s sr, no vea. bp 120-170/. bp and hr higher this pm with propofol d/c. k-4.4, hct 26.9, 1 unit prbcs infused without problems. repeat hct this eve about 10pm. resp- weaned to 5peep (40%, 25psv) with abg- 118,41,7.37,25. tv 700-800s, rr 9-12. ?wean psv tomorrow. l/s dim/coarse. cont. sx’d for small thick tan sec. via ett, clear/white orally. sats 97-98. trach care done. gi- had ogt and doboff pedi feeding tube, both seen in stomach on cxr/abd xray in am. ogt d/c’d, xrays repeated and awaiting result to verify that doboff in still in her stomach or lower. this pm, had 90cc dk. green residual, guiac positive. faint b. sounds, cont. on reglan. will restart t/f at low dose once placement confirmed as tolerated. no bm. cont. on insulin drip, bs down to 97, insulin decreased to 2units/hr and bs have been 100-122 since. see carevu also. ms- propofol weaned and d/c’d at 2:20pm, started on haldol and re’d 2 doses iv. wakes up with any tactile stimulation, occ opens eyes, ?focusing, not following commands. moves head, mouth, arms. skin- has duoderm on coccyx area, has black area on right heel. splints on. cont. on bariair bed with rotation. log rolled also with skin care.”

## Experimental Setup

We wrote all our data extraction and modeling logic in a Google Collaboratory ("Colab") notebook since this approach will allow us to integrate seamlessly with Google Cloud Platform (GCP) BigQuery (see *Data*) and share code between group members[9]. We had originally intended to use PySpark for processing the data set and feature creation. However, we encountered several speed bumps along the way. We had issues connecting to BigQuery from PySpark and getting the proper jars installed on the servers where the Colab notebook runs. We also investigated using a Google Dataproc cluster with a Jupyter notebook instead of a Colab notebook but there were issues with authenticating and connecting to the MIMIC data set. In the end, we decided to stick with BigQuery in the Colab notebook since BigQuery is itself a big data tool and it is highly compatible with the rest of the GCP ecosystem. For the NLP methods, we used the nltk and gensim libraries. For the modeling aspect of the research, we utilized the Scikit-Learn library in Python and will use Keras for any additional deep learning needs. We used the following language and package version: Python 3.7.12 and Scikit-Learn 1.0.1. In terms of hardware, the Colab notebook is run on a Python 3 Google Compute Engine back-end virtual machine and the exact specifications such as the memory change over time.

## Approach

The specific problem that we've chosen to focus on is predicting hospital readmission based on the free-form notes in the *mimiciii_notes.noteevents* table and admission and discharge dates in the *mimiciii_clinical.admissions* table. The goal of this project is to determine if a patient will be readmitted rather than a time duration between visits, so we will use a classification model to evaluate likelihood. Some aspects of the approach described below are modified from a tutorial by data scientist Andrew Long [10], but technology choices, feature engineering, and modeling techniques are all selected by the team for this project.

**Data Preparation —** A data pipeline cleans the data and prepares it for NLP processing and modeling. As mentioned in Experimental Setup, we implemented our data preparation pipeline using SQL for GCP BigQuery using the BigQuery Python client library. Each step described in the following paragraph corresponded to a new BigQuery table. We made this decision to enable us to audit each step of the transformation process and for repeatability.

To start with, we identified which of the *admissions* records are meaningful to our problem. We removed all the newborn admissions because we are concerned with patients that were admitted to the hospital and not just born. We also removed all admissions where the patient died during their hospital stay. Since we are interested in whether a patient was readmitted within a set time frame, we calculated the days until the patient's next admission for each record. In looking at the next admission date, we excluded any admissions that were considered elective since we only want to predict the unplanned re-admissions that are medically necessary. It is worth noting that the dates in this table are shifted into the distant future to ensure patient privacy but maintain the ordering of each respective event. A histogram of the counts by the number of days until re-admission (excluding nulls) can be seen in figure below (Figure 3).

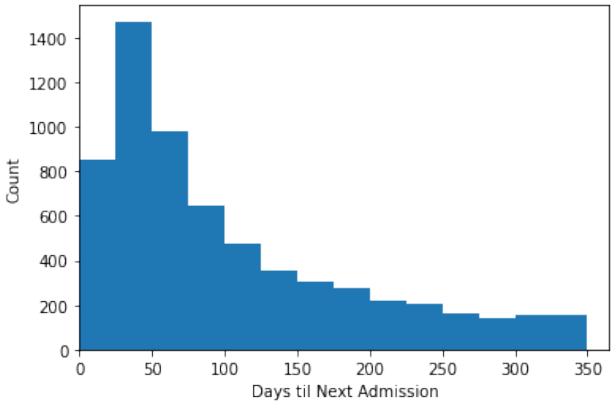


Figure 3: Histogram of the Number of Days Between Admissions

Secondly, we focused on the *noteevents* records. There are often multiple notes for a single admission for a patient of varying types. We experimented with the best approach to handle multiple notes. We tried including all note types as well as just limiting it to discharge note types. We saw slightly better model performance when only including the discharge notes. In addition, there are speed and space benefits from only including the discharge notes. We concatenated all the discharge notes associated with an admission into a single string. To standardize the notes, we converted them all to lower case and removed any punctuation except for spaces (including new line and return characters). At this point, each patient visit has a single record associated with it. Then we joined the cleaned *noteevents* and *admissions* data using the SUBJECT_ID and HADM_ID columns.

Next, we needed to identify positive and negative samples for model training. We opted for a date range of 45 days to determine if a given patient unexpectedly returned to the hospital during this time. Using the number of days until the next admission we calculated earlier, we created a target variable that is 1 if the next admission was in 45 or less days and 0 otherwise.

Finally, we split the data into training and testing data sets. We trained the model on the training data set and evaluated the model results on the testing data set. We used 70% of the data for training and the remaining 30% for testing. We calculated the number of positive (readmitted within 45 days) and negative (not readmitted within 45 days) records in the data set and found that only about 5% of all of the records are positive. This means that we have imbalanced classes in the data and a machine learning model will have difficulty distinguishing the positive records from the negative ones. To deal with this, we sub-sampled the negative class of the training data to only keep as many negative records as there are positive records. After this process, we had a balanced training data set.

**Natural Language Processing (NLP) —** Since the data we are using to predict readmission is unstructured and contains uninformative filler words, we needed to tokenize the text and generate numerical vectors that can be used to train a model. Initially, we tokenized the notes and then trained a basic doc2vec model to create the feature vector. We used the nltk library to tokenize the text which separated the text into words and ignored any words less than 2 characters long. Then we built a doc2vec model using the genism library. The doc2vec model generated representation vectors out of the words in the notes. The doc2vec model was trained over 20 epochs and then used to create the feature vector for training the model. This process was only moderately successful in terms of overall model performance.

A second NLP approach included lemmatization and resulted in overall better model performance. Ultimately, we first passed the data through WordNetLemmatizer to undergo lemmatization in order to return the lemma, or dictionary base form of each word. We then tokenized the notes using the Natural Language Toolkit (NLTK) Tokenizer to separate the text into words and ignored any words less than 2 characters long while removing a custom set of stop words and removing punctuation and numbers from the dataset. Then we built a TF-IDF Vectorized model using the TfidfVectorizer package. The model generated representation vectors out of the words in the notes. The vectorization step proved to be the most computationally expensive step taking approximately 1 hour using the hi-RAM google colab environment. This NLP approach was used in the final model.

**Modeling —** After vectorizing the notes data, we evaluated several machine learning models. All models were built using the training data set where the feature vectors were those generated by the NLP process described above and the label was whether or not the patient was re-admitted to the hospital within 45 days. We evaluated a logistic regression model, linear support vector classifier model, and a gradient boosted classifier. After evaluating the different models based on AUC, we found the best performance was with a logistic regression model. We tuned the parameters of the model and settled on a logistic regression model using the saga solver with L2 regularization and balanced class weights.

## Experimental Results

In order to evaluate different NLP approaches and machine learning classifiers to determine the most effective technique, we needed a metric to compare different models. We decided to evaluate the models using Area Under the ROC Curve (AUC), a popular metric for measuring true positives and false positives[11] along with the confusion matrix. All of the evaluation metrics were calculated against the test data set. The logistic regression model described above provided us with the best results where AUC=0.67. The confusion matrix for the model can be seen in the figure below (Figure 1). In addition, a summary of some metrics for the others types of model we evaluated can be seen in the table below (Table 2).

| | | Actual | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted | Positive | 9173 | 3370 |
| | Negative | 248 | 373 |

Table 1: Confusion Matrix for Optimized Logistic Regression Model

| Model | AUC | Accuracy | Precision | Recall |
|---|---|---|---|---|
| Logistic Regression | 0.665 | 0.725 | 0.100 | 0.601 |
| Linear SVC | 0.649 | 0.755 | 0.101 | 0.533 |
| Gradient Boosting Classifier | 0.580 | 0.904 | 0.150 | 0.222 |

Table 2: Summary of Model Results

## Discussion

Our model of lemmatization and TF-IDF vectorized model NLP approach and a tuned logistic regression model resulted in an AUC of 0.67 on the testing data. This model is a bit better than our baseline approach which was a doc2vec NLP approach and a basic logistic regression model that resulted in an AUC of 0.605 on the testing data. However, our optimized model still wouldn't be considered excellent. This less than stellar result is consistent with the findings described in the Background and Related Work section: less than 20% of published work on this dataset yielded AUC scores greater than 0.75%. Our optimized model represents a great improvement from our baseline model in terms of precision. Previously our model was predicting readmission (positive) 95% of the time but with the optimized model it is only %70 of the time. So there was a pretty significant reduction in this bias towards predicting readmission.

In addition, to the model performance metrics, we investigated which words in the clinical notes are most important for determining readmission. We calculated the importance score of each word or phrase in the TF-IDF vectorized model and ranked them to determine those that we most and least important. The two figures below (Figure 4) show the top 50 most and least important words.

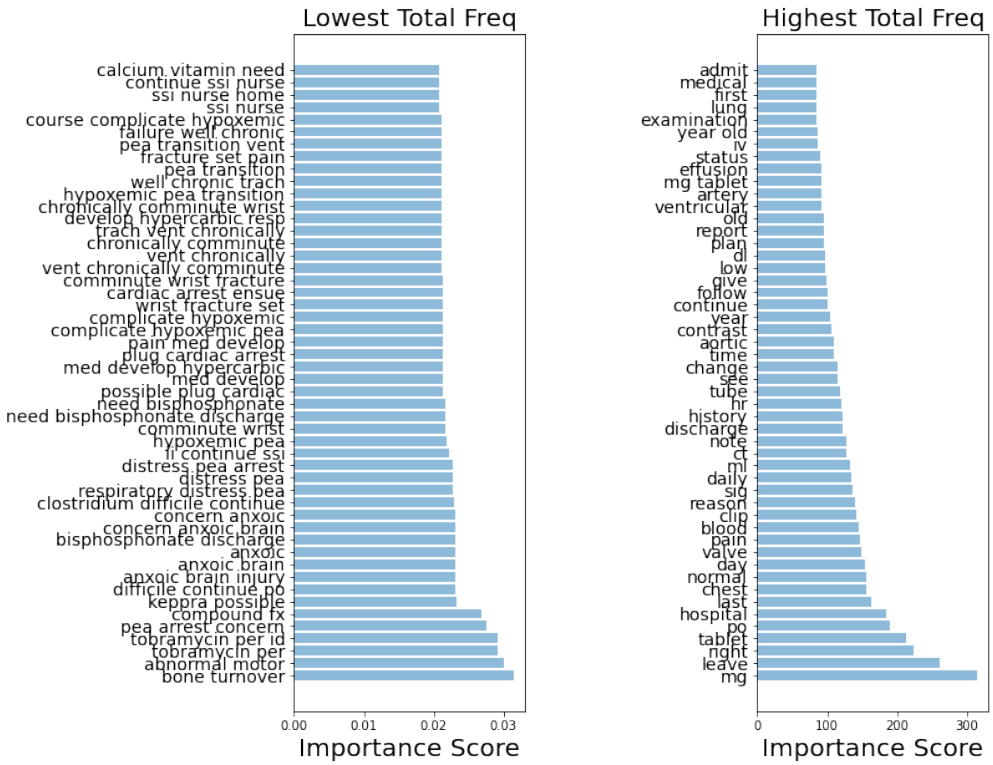


Figure 4: Word Importance

## Conclusion

In this project, we used the MIMIC-III data set with clinical notes to determine the probability of readmission for an ICU patient within 45 days. We used a data pipeline to clean the data, a TF-IDF vectorized model with lemmatization to generate features from the unstructured clinical notes, and a logistic regression model to prediction readmission. This model resulted in an AUC of 0.67 and an accuracy of 0.725. As expected, this is a challenging problem to solve and there is still lots of room for improvement. Additional model could be evaluated including a LSTM neural network and a convolution neural network. In addition, this is a computationally expensive project due to the sheer size of the clinical notes. Addition advances in NLP techniques could help improve the time it takes to vectorize the notes for analysis.

All team members contributed equally to the project.

## Supporting Information

GitHub repo: https://github.gatech.edu/etadley3/CSE6250-project

Video Presentation: https://youtu.be/neA6_CWdGTA

## References


1. Basyal, G P , Rimal, B P , Zeng, D . A Systematic Review of Natural Language Processing for Knowledge Management in Healthcare. Computer Science and Information Technology. 2020:275–285.
2. Velupillai S, Suominen H, Liakata M, Roberts A, Shah AD, Morley K, et al. Using clinical Natural Language Processing for health outcomes research: Overview and actionable suggestions for future advances. Journal of biomedical informatics. 2018;88:11-9.
3. Rajkomar A, Oren E, Chen K, Dai A, Hajaj N, Hardt M, et al. Scalable and accurate deep learning with electronic health records. npj Digital Medicine. 2018;1:18.
4. Huang K, Altosaar J, Ranganath R. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. 2019. Available from: http://arxiv.org/licenses/nonexclusive-distrib/1.0.
5. Teo K, Yong CW, Muhamad F, Mohafez H, Hasikin K, Xia K, et al. The Promise for Reducing Healthcare Cost with Predictive Model: An Analysis with Quantized Evaluation Metric on Readmission. Journal of healthcare engineering. 2021;2021:1-10.
6. Ashfaq A, Sant'Anna A, Lingman M, Nowaczyk S. Readmission prediction using deep learning on electronic health records. Journal of Biomedical Informatics. 2019;97:103256. Available from: https://www.sciencedirect.com/science/article/pii/S1532046419301753.
7. Artetxe A, Beristain A, Graña M. Predictive models for hospital readmission risk: A systematic review of methods. Computer Methods and Programs in Biomedicine. 2018;164:49-64. Available from: https://www.sciencedirect.com/science/article/pii/S0169260717313998.
8. Johnson A, Pollard T, Mark R. MIMIC-III Clinical Database (version 1.4). PhysioNet; 2016. Available from: https://doi.org/10.13026/C2XW26.
9. PySpark Google Colab: Working With PySpark in Colab; 2020. Available from: https://www.analyticsvidhya.com/blog/2020/11/a-must-read-guide-on-how-to-work-with-pyspark-on-google-colab-for-data-scientists/.
10. Long A. Introduction to Clinical Natural Language Processing: Predicting Hospital Readmission with Discharge Summaries; 2018. Available from: https://towardsdatascience.com/introduction-to-clinical-natural-language-processing-predicting-hospital-readmission-with-1736d52bc709.
11. Classification: ROC Curve and AUC — Machine Learning Crash Course. Google;. Available from: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.