Heart Failure Prediction with Natural Language Processing using Electronic Health Records

Michael Cho, MS Analytics[1]; Keith Woh, MS Analytics[2]; Samuel Yusuf, MS Analytics[3]
[1]Georgia Institute of Technology, Atlanta, Georgia

Abstract

Clinical notes are text documents created by physicians for each patient's encounter. They are typically full of information regarding the patient's medical history, medications, family history, and observations during the visit. Finding valuable information from these notes can be very important in developing approaches that would reduce a patient's risk of heart failure. Though there have been papers [Bashir et al, Kenny Ng et. al][2,6] written on using structured data about a patient to predict whether the patient would be exposed to heart failure and papers [Pakhomov et al][7] written on using NLP on EHR text to identify patients with heart failure, there have been little advances made in incorporating both structured and unstructured clinical notes to make heart failure prediction. This paper aims to provide insight on how both can be combined for better prediction. Our method creates features of most frequently occurring uni-grams, bi-grams, and tri-grams in clinical notes for case and control patients, and appends the word features to the patient's features from structured data. From these combined features, the data was fitted to several classification models, including Random Forest, Decision Trees, Neural Network, and Support Vector Machines. Using the top 3 performing models, a majority voting ensemble method was used as the final model. This model performed most optimally, with an accuracy of 72.0%.

1 Introduction

Heart Failure affects an estimated 6.5 million people in the United States, with projections showing a 46% increase in prevalence from 2012 to 2030[3]. While prediction of heart failure using structured data in Electronic Health Records (EHR) has yielded promising results, there is still a wealth of unstructured text data in EHRs to explore that can be used to improve predictions. This project aims to utilize the unstructured text data in EHRs, in addition to structured EHR data, to better predict heart failures and thus improve patient outcomes.

Several approaches have been used to analyze and predict diseases with EHR data. Kenny Ng et. al. utilized EHR records from Geisinger Clinic to predict patient Heart Failures using Logistic Regression and Random Forest models[6]. Features were extracted from a variety of structured data types, including diagnostic codes (ICD9), medication, lab, and hospitalization data. The accuracy of the Logistic Regression and Random Forest predictions was then measured as a function of prediction window days, observation window days, amount of training data, and by which data type was used (diagnostic, medication, etc.).

Serguei Pakhomov et. al. utilized EHR records from Mayo clinic to identify and predict patient Heart Failures using Natural Language Processing (NLP) and a Naive Bayes classifier[7]. In the identification step, the authors used NLP to look for synonymous diagnostic phrases within the doctor clinical notes (i.e. phrases such as heart failure, CHF [congestive heart failure], biventricular failure, and cardiopulmonary arrest) and addressing non-negated terms by excluding those terms that have negation indicators (i.e., ”no,” ”denies,” ”unlikely”) in their immediate context (7 words). For the prediction step, the authors extracted covariates from the clinical notes using Bag of Words, then predicted for Heart Failure using Naive Bayes by finding the likelihood of an outcome based on co-occurrence frequency with each of the predictive covariates.

Ananthakrishnan et al. proposed a robust EMR based model for classifying inflammatory bowel disease (IBD)[1]. The model used a combination of both structured (ICD9, prescriptions etc) and narrative data from clinical notes to produce results with higher accuracy and sensitivity. Penalized logistic regression with adaptive procedure was used to select informative variables for the final predictive model. Most papers that deal with disease prediction deal either with structured or unstructured data in insolation. This paper utilizes both forms of data to maximize information retrieved.

Bashir et al. proposed a framework to predict heart disease in a patient more accurately, where different machine learning classifiers were combined by using a majority vote based classifier ensemble[2]. The proposed prediction system used three classifiers: Naive Bayes, Decision Trees, and Support Vector Machines. This ensemble majority vote classifier outperformed individual classifiers, giving higher prediction accuracy and reliability.

Thomas et al. found that NLP was a reliable and accurate method to utilize and to extract relevant data from Electronic Medical Records[8]. The authors used data that included patients that underwent prostate biopsies. They then used an internally developed NLP program to identify which patients underwent the procedure. Although the specifics of the NLP program were not fully disclosed, the paper gives confidence that NLP can play a crucial role in extracting relevant information, allowing us to construct meaningful variables to perform prediction using Machine Learning.

James Mullenbach et al. used a convolutional neural network to predict medical codes from clinical text[4]. The authors aggregated information across the document using the convolutional neural network, then used an attention mechanism to select the most relevant segments for each of the thousands of possible codes. Although this project will not be using neural network models presented in the paper, this paper provides an insight into the current state-of-the-art method for pre-processing and extracting information out of text.

2 Method

In this section, we discuss our approach to data filtering, preprocessing the structured and unstructured data, and our process for feature generation and engineering.

Data Filtering

Our main analysis of interest is text data, which in this case are notes taken by physicians. The full dataset consists of 2,083,180 notes, with 46,146 total unique patients (10,112 case, 36,034 control). Case patients are defined as patients which have an ICD9 code that starts with '428'. Prior to doing any form of analysis, we filtered out all data that fell outside the observation window. The observation window was defined as all dates before 10 days prior to index date. For case patients, index date was the first date where the patient was diagnosed with an ICD9 code that started with '428'. For control patients, the index date was the last date of ICU admission. With the index dates calculated, we then filtered all data that occurred outside the observation period. This filter resulted in 2909 case patients and 7832 control patients. We then randomly dropped some control patients in order to achieve a more balanced 40-60% ratio for case-control patients. Our final dataset consisted of 2909 case patients, with 70927 case notes, and 4364 control patients, with 125499 control notes.

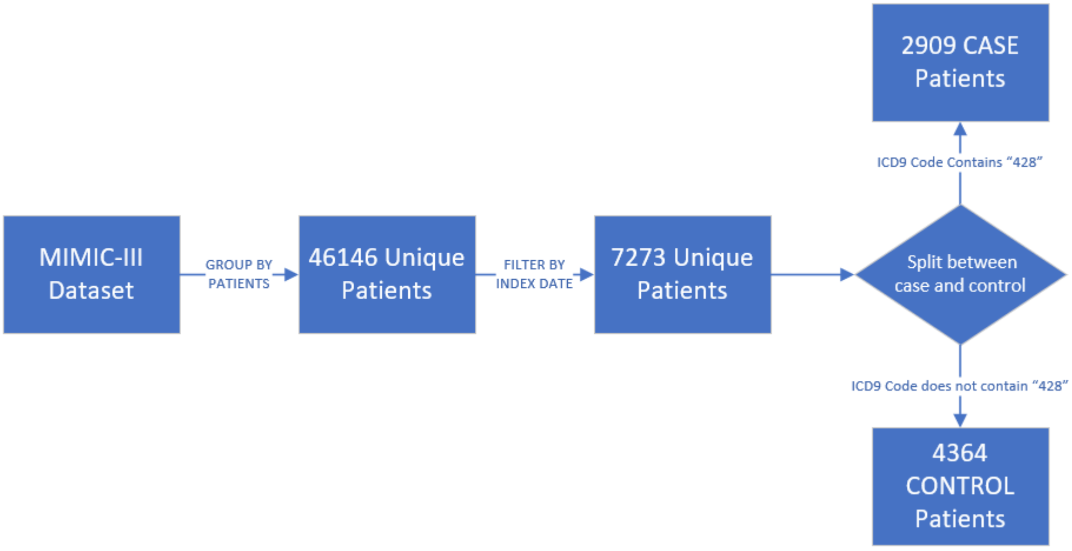

Figure 1: Diagram of Data Filtering Steps

Preprocessing Structured Data

In addition to text data, structured data was incorporated into our analysis. The data of interest for the structured data was medication, lab, diagnostic, and procedure data. These data are also filtered according to index dates. Individual patients' medication, lab, diagnostic, and procedure data were then aggregated across all visits, generating patient-level features. Medication, diagnostic, and procedure feature data were created by counting the occurrence of each event. For lab data, features were generated by taking the average lab value.

Preprocessing Unstructured Data

The unstructured data used for this paper were the clinical notes taken by physicians on the details of a patient's visit to the ICU. The data includes visits of both case and control patients. The input CSV includes twelve features of a patient's visit: ROW_ID, SUBJECT_ID (patientID), HADM_ID, CHARTDATE, CHARTTIME, STORETIME, CATEGORY, DESCRIPTION, CGID, ISERROR, TEXT, ADMITTIME. The variable of interest here was the TEXT feature of a visit. The task was to get the notes ready for feature extraction and to extract the best words as features for a Machine Learning model. First, a large text file was created with the notes for every patient. This output was then passed on to python's NLTK library to find the most frequently occurring and most sensible keywords for case patients. This process is outlined in the feature generation section below. A feature vector of these keywords was then created. Then, a text file was created for each patient that has all the notes associated with that patient. This step was done for both case and control patients. The words in the feature vector were then checked if they appear in the notes for each patient.

Notes Analysis and Features

A big component of our research was to consider the impact of physician's notes in predicting heart failure. Here we provide insights from our analysis of the text and how we created features from the notes.

For their NLP stack, Shu et al[9] primarily used cTAKES, a clinical text analysis knowledge extraction system. For working with our text data, we primarily used Python's NLTK library. The input of our analysis was 7273 text files containing the notes aggregated for 7273 case and control patients. The first step was to create an NLTK corpus containing all the text files. From there, we created a frequency distribution for the words in the corpus. The output was a python dictionary with a word as the key and the number of times the word appears in the entire corpus as the value. The next step in preprocessing was to clean up the list of words by retaining only meaningful keywords. To do this, we used the NLTK stopwords package to remove all the stop words in the English language. Upon removing stop words, there were still a lot of unrelated numbers and two-letter measurement abbreviations (mg, pt, cm) with high frequencies. These terms were also removed, leaving us with a meaningful list of words from the corpus.

The natural next step from there was to find a list of the most occurring words in the corpus and use these words as features for our classifiers. After sorting the dictionary of words in descending order of frequency, we found the top 200 words. Most of the words in the top 200 can be found in the notes for both case and control patients. This list includes 'patient', 'assessment', 'blood', 'pain', 'history', 'hospital', 'chest', and 'examination'. This makes sense as these words can appear in any patient's notes when visiting the physician. However, there were some words in the top 200 list that are unique only the case patients. These words include 'valve', 'aortic', 'ventricular', 'pulmonary', 'artery', and 'rhythm', and are all words that are directly related to or have some indirect relationship to heart failure. This is good news because it hints that our algorithm would be able to learn which keywords are related to patient's encountering heart failure. With that insight on hand, we concluded on using these top 200 words as features for our experiment.

Previous research had taken advantage of extracting a number of n-gram features other than uni-grams[5]. Therefore, we also experimented with using our NLTK framework described above to get the top 100 bi-grams and top 50 tri-gram features to add to our note features. N-grams beyond tri-grams were not used, as there was also research showing that n-grams greater than three are not as effective predictors in a model[5].

Finally, to get the notes features ready for modeling and experiments, we created a Conditional Frequency Distribution which outputs a table showing the number of times the selected uni-gram, bi-gram, and tri-gram features appear in the notes for each patient. Each column in the table is a feature corresponding to one n-gram created in the previous steps, while each row corresponds to a patient.

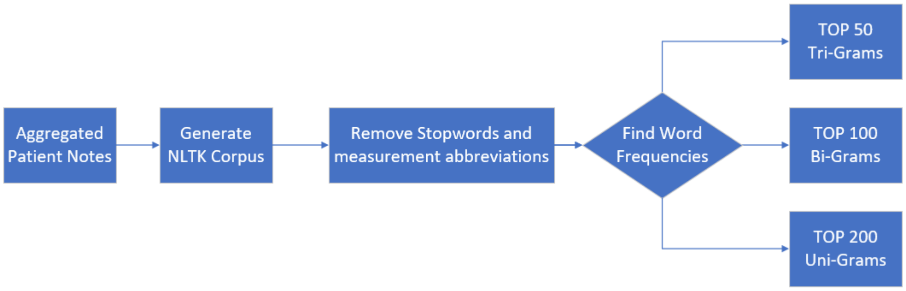

Figure 2: Diagram of Note Analysis Steps

Feature Construction for Machine Learning Models

Using Spark, all features were combined from structured and unstructured data at the patient-level. A svmlight file format was then generated for use in our prediction modeling step.

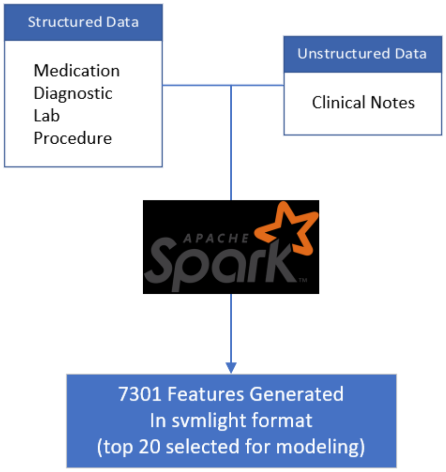

Figure 3: Diagram of Feature Construction Steps

3 Experimental Results

The next task at hand was to do predictive modeling. Our predictive modeling was done with python's scikit-learn package, as well as pyTorch for Neural Networks. With the patient svmlight file as input, we ran different classifiers on our data and compared performance. First, we converted the input into an array. Then we split our data into training, validation, and testing sets, where each set accounted for 56%, 14%, and 30% of the data, respectively.

The SVM L1 penalty classifier had a 69.60% accuracy and 74.34% AUC score on the validation set. This was a slight improvement in accuracy from a regular SVM classifier without regularization which had accuracy of 61.1% and AUC of 64.2%. Next, we fit a Decision Tree classifier which had a 65.6% accuracy on the validation set. We then fit a Random forest model, which had an accuracy of 70.4%.

Finally, we fit a Feedforward Neural Network on our data. After experimenting with multiple hyperparameters, our best model was one with 4 hidden layers, 256 nodes in each layer, each layer fully connected with ReLU activation, and SGD optimization with 0.001 learning rate and 0.9 momentum. This was the chosen model despite not having the highest accuracy at 68.2% because it had the highest recall at 69.0% and AUC at 74.8%.

The table below shows the classification metrics on the validation set:

	Accuracy	AUC	Precision	Recall	F1 Score
SVM	61.10%	64.20%	56.31%	53.86%	55.06%
SVM L1	69.90%	74.34%	**68.00%**	53.38%	**59.81%**
Decision Tree	65.60%	X	59.13%	**54.50%**	56.72%
Random Forest	70.40%	X	71.13%	47.87%	47.87%

Table 1: Performance of Classification Models (Best Performances in Bold)

The table below shows the performance for some of the Feedforward Neural Network models:

Neural Network			Accuracy	AUC	Precision	Recall
Hidden Layers	Nodes in Each Layer	Learning Rate				
2	256	0.001	0.672	0.720	0.658	0.434
3	256	0.001	**0.691**	0.734	**0.684**	0.472
4	256	0.001	0.682	**0.748**	0.601	**0.690**
4	512	0.001	0.690	0.742	0.634	0.595

Table 2: Performance of Feedforward Neural Network Models (Best Performances in Bold)

The confusion matrix for the all the models on the validation set is displayed below:

	Predicted: Control	Predicted: Case
Actual: Control	431	173
Actual: Case	191	223

Table 3: Linear SVM Confusion Matrix

	Predicted: Control	Predicted: Case
Actual: Control	500	104
Actual: Case	193	221

Table 4: Linear SVM Confusion Matrix with L1 Regularization

	Predicted: Control	Predicted: Case
Actual: Control	438	159
Actual: Case	192	230

Table 5: Decision Tree Confusion Matrix

	Predicted: Control	Predicted: Case
Actual: Control	515	82
Actual: Case	220	202

Table 6: Random Forest Confusion Matrix

	Predicted: Control	Predicted: Case
Actual: Control	404	193
Actual: Case	131	291

Table 7: FeedForward Neural Network Confusion Matrix

After much experimentation, we decided to do an ensemble model with the following classifiers: linear SVM with L1 penalty, Random Forest, and a Feedforward Neural Network. The reason for choosing these classifiers was because they were robust and contain natural remedies to the curse of dimensionality. SVM with L1 penalty would not overfit because of the excessive regularization that occurs. Random forest uses a collection of decision trees to make prediction and an individual tree only uses a subset of the features. Finally, a Neural Network's performance and ability to generalize is also not as affected by the curse of dimensionality as most other models. Results on the ensemble model are below (note that results were based on our final test set):

	Predicted: Control	Predicted: Case
Actual: Control	1145	186
Actual: Case	424	427

Table 8: Ensemble Model Confusion Matrix

The table below displays the top 20 features with the highest coefficients from the SVM L1 model:

Feature Type	**Feature Name**
Structured	proc99291
Unstructured	textmoderate
Structured	medtuss51
Unstructured	textventricular
Structured	meddesi10
Structured	diag9351
Structured	diag4255
Structured	diag1573
Structured	proc32020
Unstructured	textcompared
Structured	diag42971
Unstructured	textvalve
Structured	diag25082
Structured	lab51006
Structured	medenal5
Structured	proc99261
Structured	proc99232
Structured	proc99239
Structured	diag3970
Structured	medntg100pb

Table 9: Top 20 Features Extracted

From the list of the top features above (shown in descending order), most of the significant features come from fields in our structured data set: they are procedure codes, diagnostic codes, and medication codes. However, four features are from the notes: 'moderate', 'ventricular', 'compared', and 'valve'. These are words that are found primarily with the notes of case patients.

4 Discussion

Despite a relatively high accuracy of 72% from our final ensemble model, we note that the recall for the model is relatively low. This is of concern to us, because in the medical context, we are more interested in identifying True Positives since those are the patients who will have heart failure. Moving forward, there is definitely a need to build models that will be able to achieve higher recall scores. Out of all the models we have experimented with, Neural Network obtained the highest recall, and is a good direction for future work. We can also explore alternative NLP techniques, such as word2vec, and using more complex models, such as Convolutional Neural Networks (CNN), to deal with the higher dimensional data. Time sequence of patient visits could also be accounted for, providing a richer set of information for model building.

5 Conclusion

The goal of this paper was to present a novel approach for predicting heart failure using a combination of structured and unstructured features from Electronic Health Record data. Our approach utilized the 200 most commonly occurring words (uni-grams), top 100 bi-grams, and top 50 tri-grams from physician clinical notes, in conjunction with more structured patient information, which included medication, lab, diagnostic, and procedure data. Using the combined set of structured and unstructured features, the data was fit to multiple different classification models, including Random Forest, Decision Trees, Neural Network, and Support Vector Machines. Using the top 3 performing models, we then used a majority voting ensemble method as our final model. This model performed most optimally, with an accuracy of 72.0%. Although the initial results are promising, more experimentation is required. Future work should be dedicated toward improvement of recall performance, exploring alternative NLP techniques, and incorporating the time sequence of patient visits.

References

1. Ananthakrishnan AN, et al. Improving Case Definition of Crohn's Disease and Ulcerative Colitis in Electronic Medical Records Using Natural Language Processing: A Novel Informatics Approach. Inflamm Bowel Dis. 2013 June; 19(7): 1411-1420.
2. Bashir S, Qamar U, Javed MY. An Ensemble based Decision Support Framework for Intelligent Heart Disease Diagnosis. International Conference on Information Society. 2014; 38(4): 259-264
3. Benjamin EJ, et al. Heart Disease and Stroke Statistics 2017 Update [Internet]. American Heart Association, Inc. 2017; 135:e379. Available from:
circ.ahajournals.org/content/circulationaha/early/2107/01/25/CIR.0000000000000485.full.pdf
4. Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J. Explainable Prediction of Medical Codes from Clinical Text [Internet]. 2018 Feb. Available from: https://arxiv.org/pdf/1802.05695.pdf
5. Furnkranz J. 1998. A Study Using n-gram Features for Text Categorization. Austrian Institute for Artificial Intelligence. Technical Report OEFAI-TR-98- 30.
6. Ng K, Steinhubl SR, deFilippi C, Dey S, Stewart WF. Early Detection of Heart Failure Using Electronic Health Records: Practical Implications for Time before Diagnosis, Data Diversity, Data Quantity and Data Density. Circ Cardiovasc Qual Outcomes. 2016 Nov; 9(6): 649-658.
7. Pakhomov S, Weston SA, Jacobsen SJ, Chute CG, Meverden R, Roger VL. Electronic Medical Records for Clinical Research: Application to the Identification of Heart Failure. The American Journal of Managed Care. 2007 Jun; 13(6): 281-288.
8. Shu D, R Kannan M, Siddhartha J. Using Natural Language Processing to Screen Patients with Active Heart Failure: An Exploration for Hospital-wide Surveillance [Internet]. Division of Health and Biomedical Informatics, Northwestern University. Available from: https://arxiv.org/pdf/1609.01580.pdf
9. Thomas AA, Zheng C, Jung H, Chang A, Kim B, Gelfond J, Slezak J, Porter K, Jacobsen SJ, Chien GW. Extracting data from electronic medical records: validation of a natural language processing program to assess prostate biopsy results. World J Urol. 2014 Feb; 32(1): 99-103.