# Hospital Readmission Prediction Based on Medical Notes

Wenjie Yao, Shenghua Xiang, Xinge Zhao
**Georgia Institute of Technology, Atlanta, Georgia, US**

## Abstract

*Excessive hospital readmissions are causing significant financial and health damage to affected individuals and communities. Many such readmissions can be avoided through effective actions, such as making sure that patients leave the hospital with appropriate medications, clear instructions for follow-up care etc. In order to enable doctors and patients to make more well-informed instructions and decisions after discharge, we identified readmission events and their corresponding discharge notes from over 2 million records and developed prediction models on readmissions using machine learning algorithms and natural language processing techniques like TF-IDF and Word2vec. Our model reports a true positive rate of up to 71.6% in prediction, which can serve as an effective indicator for possible future readmissions.*

Keywords: Hospital readmission, predictive modeling, medical notes; NLP; TF-IDF; Word2vec

Narration PPT Edition: https://drive.google.com/open?id=1KHUL4IAvGPdsEGpa4qs08ejZO-xNsz6_

## 1. Introduction

Avoidable readmissions have been considered not only as a healthcare quality measure, but also as a means to reduce healthcare cost. According to previous studies, nearly 20% of all Medicare discharges had a readmission within 30 days [1]. The Medicare Payment Advisory Commission (MedPAC) has estimated that 12% of readmissions are potentially avoidable, which could save at least $1 billion every year for Medicare system [2]. To tackle the problem, The Affordable Care Act (ACA) established the Hospital Readmission Reduction Program (HRRP) in 2012. Under this program, hospitals are financially penalized if they have higher than expected risk-standardized 30-day readmission rates for certain diseases [3]. To help hospitals and patients better prevent readmissions, we developed predictive models based on previous medical notes, tackling three major challenges:

**Challenge 1: Readmission identification.** We acquired the MIMIC-III critical care database [4], which include over 2 million medical notes and nearly 60,000 admission records. Thus, cleaning and merging the datasets to identify readmissions within 30 days of initial discharge required significant effort. Using R, we found 59,652 admission events and their corresponding medical notes in total, out of which 3,380 are characterized as readmission.

**Challenge 2: Feature extraction.** In order to transform medical notes to input features, first we used Zeppelin to transform the texts to a cleaner format by transforming them to lowercase, removing symbols and numbers. Then we applied TF-IDF to the large text corpus, ran feature selection and saved the label-feature pairs in SVMLight format. On the basis of TF-IDF, we explored Word2vec technique which utilizes neural networks and provides a more semantic sensitive means for extracting features.

**Challenge 3: Supervised readmission classification.** Then, we built predictive models of readmission using machine learning approaches, including Logistic Regression, Support Vector Machine (SVM), Random Forest and Feedforward Neural Network. We experimented with various parameter tuning techniques to achieve better performance, our best model, Feedforward Neural Network, achieves a true positive rates (TPRs) up to 71.6% (in predicting readmission) at a false positive rate(FPR) of 28%, A cross-validation AUC of 0.76 and test AUC also of 0.76.

## 2. Related Work

Hospital readmissions and their effects in relation to many aspects have been widely studied, such as patient mortality [5] and congestive heart failure [6]. Traditional prediction models on readmissions usually rely on institution-specific patient-level factors and basic statistical methods [7]. However, prediction using machine learning methods and medical notes have drawn less attention, despite its obvious potential and importance.

One study used natural language processing tools to analyze and assign a sentiment score for each hospital admission and discharge notes and found that sentiment measured in hospital discharge notes is associated with readmission and mortality risk [8]. Another study developed a regression risk score model based on administrative data to predict hospital readmissions for heart failure [9]. A more methodologically rigorous approach, as seen in Kansagara D. Englander's work [10], gathers more than 20 factors related to patients and hospitals, and runs a thorough factor analysis to find associations for readmission.

Extracting features from medical notes would inevitably generate high dimensional feature vector space because of the large vocabulary used, also correlations between features may further increase the redundancy. Thus, dimensional reduction is critical for computing efficiency and model performance. In this study [11], the researchers utilized drug domain ontology to reduce redundancy by clustering concepts and revealing their functionality, they proposed a feature reduction method for high dimensional dataset to predict the probability of 30-Day heart failure readmission. In addition, Radovanovic S. Vukicevic developed a feature selection method that exploits hierarchical domain knowledge together with data, and the method is evaluated on predicting 30-day hospital readmission for pediatric patients from California [12].

Word2vec maps each word in the lexicon to a vector space while similar words are closer in distance. Lilleberg J. shows that tf-idf and word2vec combined can outperform tf-idf because word2vec provides complementary features (e.g. semantics that tf-idf can't capture) to tf-idf [13].

## 3. Data and Environment Description

### 3.1 Data Sources

We acquired data from MIMIC-III (Medical Information Mart for Intensive Care III) critical care database [4], which is a large, freely-available database comprising de-identified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012.

The two tables we used are admissions and note events.

| Table Name | Description |
|---|---|
| ADMISSIONS.csv | This table records over 110 thousand patients' hospital admission events. Each event is identified with unique hospital ID and patient ID. Available information includes admission and discharge dates, demographic information, the source of the admission and so on. |
| NOTEEVETS.csv | This table contains 2 million medical notes for patients. |

Table 1: Data Sources Summary

## 3.2 Data Joining

**Assign labels for each record in admissions table.** First group the table by patients, for each patient, sort the records by discharge time, if the time interval between discharge and admission is less the 30 days, then we would label this record as 1 otherwise 0.

**Join admissions and noteevents table.** Select all notes with category "Discharge summary", inner join them with admission table on patient ids and hospital ids. Then filter the data by selecting records who were written between admission and discharge date. For a single admission, there may be several discharge notes. Merge these notes to a single one. The final measures can be seen in Table 2.

| Number of unique patient. | Number of readmission records | Number of records |
|---|---|---|
| 46520 | 3939 | 58976 |

Table 2: General Statistics After Joining

## 3.3 Data Cleaning

After joining two datasets together, significant data cleaning is still needed. The bulk of the data cleaning process involved transforming the texts into lowercase, removing numbers and various symbols.

We deployed Zeppelin notebook on Spark with 4 clusters for interactive data processing. We utilized Scala dataframe functions to complete these tasks and then saved the processed texts and labels to csv files.

```
zero_set_transformed.show
```

```
+---+--------------------+
| _1|                  _2|
+---+--------------------+
|  0|admission date   ...|
|  0| admission date  ...|
|  0| admission date  ...|
|  0|admission date   ...|
|  0|admission date   ...|
|  0|admission date   ...|
|  0|admission date   ...|
|  0|name      known l...|
|  0|admission date   ...|
|  0|admission date   ...|
|  0|admission date   ...|
|  0|admission date   ...|
```

Figure 1: label and texts after data cleaning

## 3.4 Overview

Figure 2 is a flowchart of our data processing and feature construction process for a more clear and intuitive representation purpose.

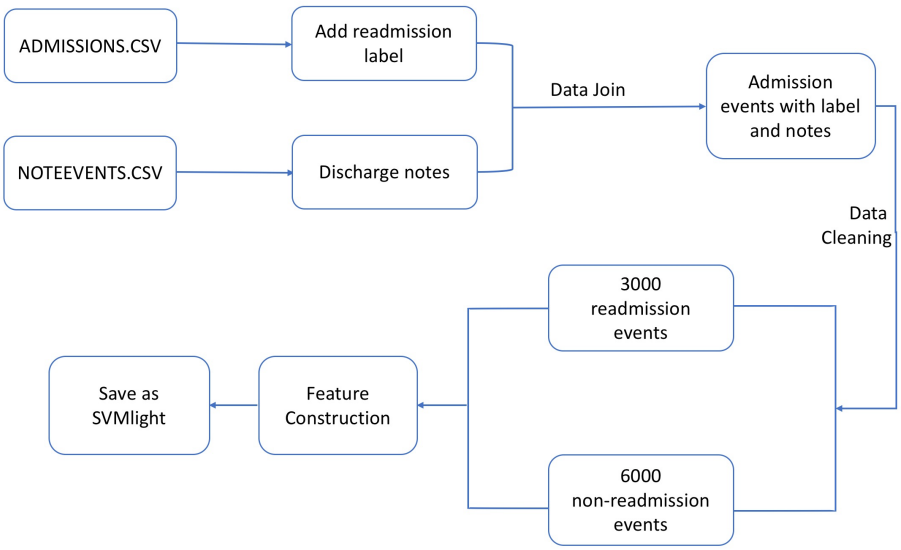


Figure 2: Data Processing and Feature Construction Flowchart

## 3.5 Environment Setup

**Hardware:** Ubuntu 16.04 machine with 4GB RAM and 2 Cores.

**Software:** Zeppelin on Apache Spark interpreter with 4 clusters.

Python 2.7 with Scikit-learn, Matplotlib, Numpy, Pandas, Pytorch libraries.

R version 3.3.2

# 4. Predictive Model of Readmission

## 4.1 Feature Selection

After data cleaning, we loaded the csv files into Python, since we only have 3939 readmission records, it's ideal to select 6,000 non-readmission records randomly. Together we now have almost 10,000 records in total.

Then we applied TF-IDF to the text column which results in a sparse matrix of 9693 rows and 73670 columns. Here we used univariate selection to select 4,000 features.

Next, apply Word2vec to the text columns. With 7 words context window, using 2000 dimensions vector to represent every word, then for each text, we sum up every word vector to represent the text. Thus, Word2vec gives us 2000 features.

Inspired by Lilleberg J. et. al. [13], we decided to combine features from TF-IDF and Word2vec to form new features for each label. We each selected 2,000 features from TF-IDF and Word2vec, now we have a new combined feature of 4,000.

After feature selections, we saved the label and corresponding features to SVMLight format as further machine learning algorithm input.

## 4.3 Evaluation of the Models

To effectively validate our model, we split the whole dataset into 70% training set and 30% testing set. We experimented with Logistic Regression, SVM and Random Forests first. On training set, we used grid search with 10-fold cross validation to select best parameters, we also used 30% of the training set as validation set to tune a single parameter, as seen in Figure 3, in which the best C for Logistic Regression is 22.12 since it maxed the accuracy on validation set. We then trained the models using best parameters and generated predictions on the testing set.

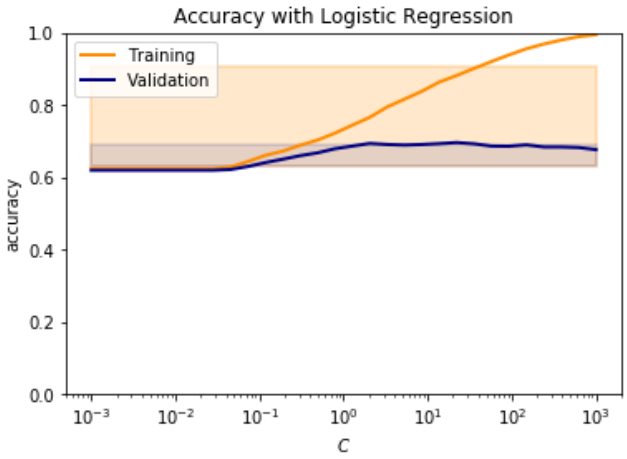


Figure 3: Tuning C on Logistic Regression with TF-IDF features

First, we experimented with TF-IDF features, results can be seen in Table 3. Surprisingly, logistic regression with simply parameter tuning outperforms other models. With a testing AUC of 0.75, and testing accuracy of 0.7.

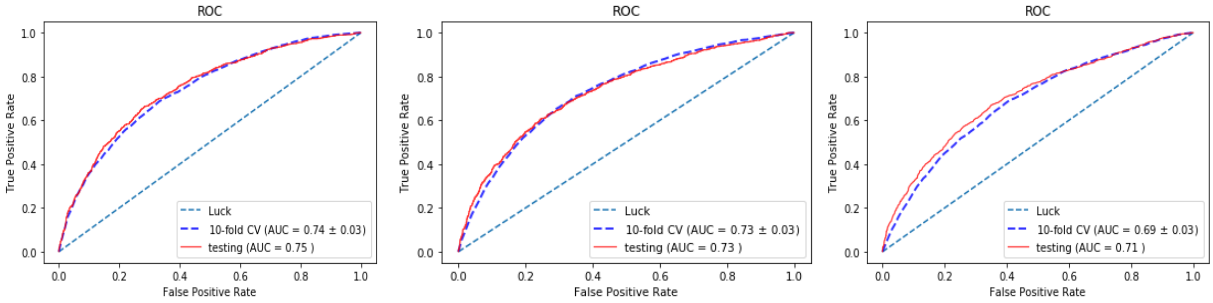


Figure 4: ROC curves of Logistic Regression, SVM and Random Forests using TF-IDF features

| | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Logistic Regression | 0.705 | 0.661 | 0.493 | 0.564 |
| SVM | 0.698 | 0.668 | 0.438 | 0.529 |
| Random Forests | 0.676 | 0.668 | 0.325 | 0.438 |

Table 3: Metrics using TF-IDF features

Second, we applied features acquired from Word2vec. The original word2vec model transforms each word into a vector. We sum up vectors of words to represent a text corpus. However, the features constructed by summing up word vectors performed poorly, which indicates summing up vectors is not a good way for exploring semantic information in this case. Figure 4 is an example of logistic regression ROC curve using such construction method.

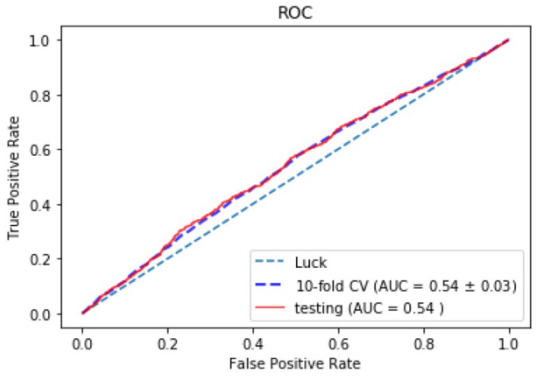


Figure 5: ROC curves of Logistic Regression using Word2vec features

| | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Logistic Regression | 0.612 | 0 | 0 | 0 |
| SVM | 0.593 | 0 | 0 | 0 |
| Random Forests | 0.554 | 0 | 0 | 0 |

Table 4: Metrics using Word2vec features

Third, we experimented with combined features from TF-IDF and word2vec. However, the overall performance was close to using just TF-IDF features, but didn't improve at all, as seen in Figure 4 and Table 4, since Word2vec features performed much worse than TF-IDF features.

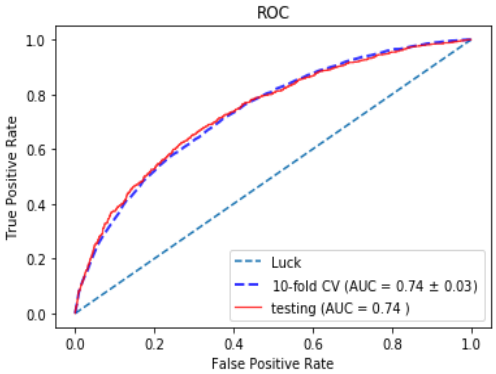


Figure 6: ROC curves of Logistic Regression using combined features

| | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Logistic Regression | 0.697 | 0.661 | 0.493 | 0.564 |
| SVM | 0.663 | 0.624 | 0.415 | 0.503 |
| Random Forests | 0.647 | 0.619 | 0.232 | 0.338 |

Table 5: Metrics using Combined features

**Further experiment with Feedforward neural network:**

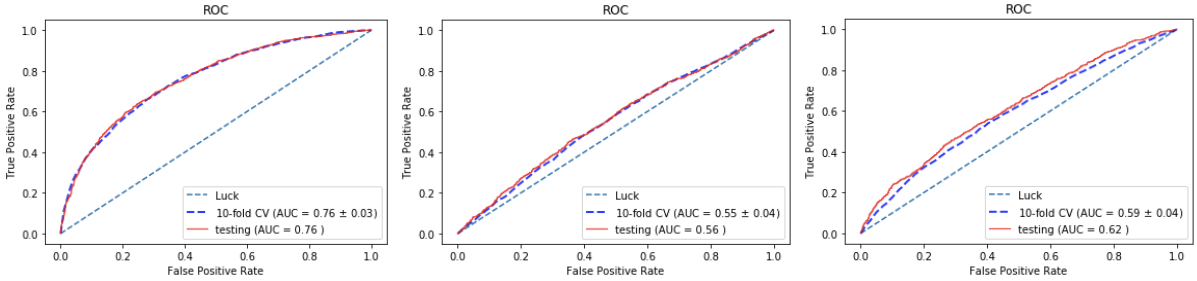


Figure 7: ROC curves of Feed Forward Neural Network with TF-IDF, word2vec, combined features

Feedforward neural network is robust to overfitting since it does not possess too much hypothesis. We trained and test our data on a 2-hidden-layer Feedforward neural network, with 250 hidden unit for each hidden layer.

From Figure 5, We can see that neural network works well with TF-IDF features, which has a true positive rates (TPRs) up to 71.6% (in predicting readmission) at a false positive rate(FPR) of 28%, also it gives that best AUC of 0.76 so far. It performs poorly with summed-up word vectors as expected. Also, our result shows that, Feedforward neural network is dominated by word2vec features when these two features combined, Since TF-IDF features are quite sparse, while word2vec features are dense, and Neural Networks are more sensitive to the sparsity of the training data, compared to Logistic Regression, SVM and Random Forest.

**Best Model performance:**

Our best performed model so far is Feedforward Neural Network with TF-IDF features shown in Figure 5. All the results are averaged over 10 trials. The most important metric in this case is the true positive rate (TPR), i.e., how many readmissions were correctly predicted as positive in our model. Our best model was able to predict 71.6% of the readmissions in testing at a false positive rate (FPR) of 28%, which is deemed useful for hospitals and patients. At the same time, a high FPR facilitates more inspections of possible avoidable readmissions, which is also beneficial. In practice, doctors and patients can adjust the TPR/FPR ratio to match their risk adversity. Considering how less readmissions occur; these results are much more predictive than guessing by chance.

## 4.4 Further Discussion of the Models

In this section, we discuss some insights we obtained while conducting the experiments.

First, due to the limitation of the data, we are unable to distinguish planned readmissions from unplanned readmissions, also for unplanned readmissions, some of them may not be preventable like accidents. Acquiring data that includes more specific categories will help to resolve this problem. In addition, we only defined readmission at a 30-day time interval after the initial discharge, which may be too narrow for preventable readmissions over longer periods like 90 days.

Second, since we built the model from existing notes, the lexicon in our model is limited. And considering computing power and efficiency, our model was built on 4,000 most relevant word features, however a typical medical dictionary contains more than 100,000 entries, which limits our prediction capability on a more general case.

In the last, due to time limitations, we were only able to test with TF-IDF and Word2vec in constructing features and summed-up word vectors as features in Word2vec didn't perform well. There are more advanced and semantic sensitive ways of extracting features from texts. Our model still has a large space for improvement.

## 5. Conclusions and Future Work

Due to the large financial and medical expenditures caused by avoidable readmissions every year, there is a need for a data-driven system to better help doctors and patients make more well-informed instructions and decisions. In this paper, we provided a framework to extract features from medical notes and make predictions on readmission.

Our work first provides a clear process for joining admission records and their previous discharge notes. We were able to identify 3939 readmissions out of 58976 admission records. We next presented methods for extracting and selecting features using TF-IDF, word2vec separately and combined, then we experimented with different machine learning approaches to achieve the best performance possible.

Our models used 9939 records in total. Specifically, at a false positive rate of 28%, the Feedforward neural network model was able to predict 71.6% of the total readmission correctly. In addition, even the false positives provided valuable insight since they represent records with high risks of future readmission.

**Future Research Directions.** Future research should seek to further categorize the data as unplanned and planned readmission, also further expand and validate our model with more advanced sentiment-related feature extraction approaches like ontology domain knowledge. In addition, other factors such as drug use, patient satisfaction etc. can also be added as prediction features. Other work would be building an app or a website tool to put the models into real practice.

## 6. References


[1] Jencks SF, Williams MV, Coleman EA. Rehospitalizations among patients in the Medicare fee-for-service program. N Engl J Med. 2009; 360:1418–1428.

[2] Medicare Payment Advisory Commission. Report to the Congress: promoting greater efficiency in Medicare. [Accessed May 25, 2014]; http://www.medpac.gov/documents/jun07_entirereport.pdf

[3] McIlvennan, C. K., Eapen, Z. J., & Allen, L. A. (2015). Hospital Readmissions Reduction Program. Circulation, 131(20), 1796–1803. http://doi.org/10.1161/CIRCULATIONAHA.114.010270

[4] MIMIC-III, a freely accessible critical care database. Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. Scientific Data (2016). DOI: 10.1038/sdata.2016.35. Available from: http://www.nature.com/articles/sdata201635

[5] Diya, L., Heede, K. V., Sermeus, W., & Lesaffre, E. (2011). The relationship between in-hospital mortality, readmission into the intensive care nursing unit and/or operating theatre and nurse staffing levels. Journal of Advanced Nursing, 68(5), 1073-1081. doi:10.1111/j.1365-2648.2011.05812.x

[6] Vinson, J. M., Rich, M. W., Sperry, J. C., Shah, A. S., & McNamara, T. (2015, April 27). Early Readmission of Elderly Patients With Congestive Heart Failure. Retrieved from https://onlinelibrary.wiley.com/doi/full/10.1111/j.1532-5415.1990.tb03450.x

[7] Hasan, O., Meltzer, D.O., Shaykevich, S.A. et al. J GEN INTERN MED (2010) 25: 211. https://doi.org/10.1007/s11606-009-1196-1

[8] McCoy TH, Castro VM, Cagan A, Roberson AM, Kohane IS, Perlis RH (2015) Sentiment Measured in Hospital Discharge Notes Is Associated with Readmission and Mortality Risk: An Electronic Health Record Study. PLoS ONE 10(8): e0136341. https://doi.org/10.1371/journal.pone.0136341

[9] Philbin, E. F., & Disalvo, T. G. (1999). Prediction of hospital readmission for heart failure: Development of a simple risk score based on administrative data. Journal of the American College of Cardiology, 33(6), 1560-1566. doi:10.1016/s0735-1097(99)00059-5

[10] Kansagara, D., Englander, H., Salanitro, A., Kagen, D., Theobald, C., Freeman, M., & Kripalani, S. (2011). Risk Prediction Models for Hospital Readmission. Jama, 306(15), 1688. doi:10.1001/jama.2011.1515

[11] Lu, S., Ye, Y., Tsui, R., Su, H., Rexit, R., Wesaratchakit, S., . . . Hwa, R. (2013). Domain Ontology-based Feature Reduction for High Dimensional Drug Data and its Application to 30-Day Heart Failure Readmission

Prediction. Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing. doi: 10.4108/icst.collaboratecom.2013.254124
[12] Radovanovic, S., Vukicevic, M., Kovacevic, A., Stiglic, G., & Obradovic, Z. (2015). Domain knowledge Based Hierarchical Feature Selection for 30-Day Hospital Readmission Prediction. Artificial Intelligence in Medicine Lecture Notes in Computer Science, 96-100. doi:10.1007/978-3-319-19551-3_11
[13] Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and Word2vec for text classification with semantic features. 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC). doi:10.1109/icci-cc.2015.725937