Performance Analysis of Machine Learning Algorithms for Predicting Hospital Readmissions

Rajasekhar Reddy Mekala
Department of Computer Science
University of California, Irvine
rmekala@uci.edu

Agniraj Baikani
Department of Computer Science
University of California, Irvine
abaikani@uci.edu

Shravan Balamurugan
Department of Computer Science
University of California, Irvine
shravanb@uci.edu

Abstract

In this project, we plan to analyze the problem of predicting hospital readmission rates among diabetic patients using the "Diabetes 130-US hospitals" dataset. Traditionally, this problem is dealt with by using statistical machine learning algorithms like Naive Bayes, K-Nearest Neighbors, and Logistic regression. These algorithms are known to not perform well on non-separable and high-dimensional datasets. To overcome these pitfalls, we will explore advanced techniques such as random forests, ensemble methods, and neural networks. Missing data, overfitting, and feature engineering are some of the challenges that we will encounter. The ideal outcome of the project would be to gain deeper insights into hospital readmission rates and investigate robust methods that can make improved predictions than the statistical methods. Our experiments show that Random forests performed better than other methods in the predictions.Attributes like gender, race, total number of medications, lab procedures, admission type, time in hospital of the patient had a significant influence in these predictions.

1 Introduction

Diabetes is chronic and one of the most prevalent diseases in the United States and it is expected to be the 7th biggest mortality factor by 2030[1]. Hence, there has been a significant improvement in collecting medical data across hospitals, to provide quality care and personalized treatment[2]. Assessing the factors contributing to the disease is critical and one of them is to understand the patient's readmission rates. Studies suggest late readmissions pose huge financial problems for developing countries[3]. Predicting readmission rates is not only useful for providing cost-efficient treatments and detecting potential problems in the early stages but also helps to evaluate the judgment of practitioners and hospitals. Many studies [4],[5] show that readmission after 30 days cannot be attributed to the previous treatment and could be influenced by many external factors.

So, our work aims to understand the key reasons and trends in diabetic readmissions in US hospitals and propose robust prediction models using machine learning techniques for the 30-day window period of readmissions.

2 Data

We use the "Health Facts National Database", which consists of 101766 encounters of patients admitted in US hospitals for diabetic treatments. The patients were treated in-hospital between 1-14 days and it contains 50 features(37 nominal and 13 numeric), like records of the prescribed medication, laboratory tests, gender, age, and many others which are discussed in detail in section 2.1. Each record contains information, if a patient was readmitted >30 days, <30 days, or not readmitted at all. A detailed description of all the attributes is provided in Table 1 of Strack et.al. We consider all readmissions after 30 days to be independent of the previous admissions.

2.1 Data preprocessing

In our exploration, we observed that some columns of the tabular dataset contain missing data. Figure 1 shows the connection between various encounters with missing data. We ignore weight, medical specialty, and payer code columns as they would not provide much information since most of the data with these columns is missing. We dropped encounter ids and patient numbers as we felt they should be irrelevant to the patient's readmission rates. Dosages citoglipton, examide were also ignored as they were not administered to any of the patients. We chose not to consider multiple encounters with the same patient for our study as such interactions cannot be considered independent encounters. Figure 2 shows that race is a significant factor in predicting readmission. So, Missing race values were replaced with a new generic field.

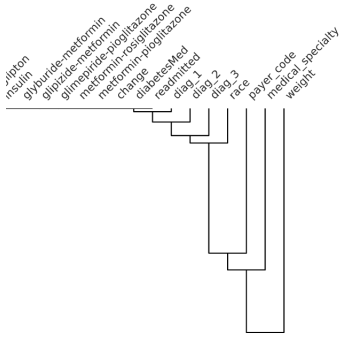


Figure 1: Dendogram of missing values.

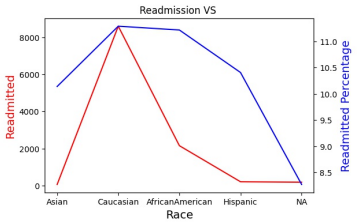


Figure 2: Readmissions (absolute and percentages) plotted for different races

Age column is nominal in the dataset, consisting of age groups in ranges of 10 years ([0-10] - [90-100]) which were replaced with their mean and converted to a numeric column. We noticed 3 rows with unknown gender and decided to leave them. All the drug dosages consisted of 4 possible values {Up, Steady, Down, No}. We treated that having steady dosage or no dosage as the same condition (no effect), change in dosage as a negative signal, and converted all of them to numeric columns. Similarly, columns containing data related to change in dosage, change in diabetes medications, maximum serum dosage, and A1C test result were converted to numeric fields from nominal values. We also noticed that although columns like admission type id and admission source id and discharge disposition id were numerical, they were not in any particular order of importance. So, we ignored redundant ids and clubbed a few other ids together to make sense of the data in terms of emergency admissions.

As mentioned previously, the dataset consisted of readmission rates classified as readmitted in <30 days, >30 days, and not readmitted at all. We tried to predict only readmissions within 30 days of the initial admission. After preprocessing the readmitted column, we noticed the data is heavily skewed as there was an imbalance with readmissions shown in Table 1. To tackle this, we used the SMOTE (Synthetic Minority Oversampling Technique), RandomOverSampler sampling techniques to synthetically generate readmission encounters while training. As explained later in the results section, this technique significantly improved the classification performance.

Readmitted	before SMOTE	after SMOTE
No	51303 (91.1%)	51303 (50%)
Yes	5041 (8.9%)	51303 (50%)

Table 1: Data points before and after oversampling

2.2 Feature engineering

After analyzing the correlations across of the dataset, we find that the categorical features diag1, diag2, and diag3 have many unique values. One-hot encoding of these fields would result in too many input features causing dimensionality problems. So, we use the domain knowledge tables from Strack et.al as a reference and reduced the unique count from around 700's to 10 categories for these three variables. We have also replaced the missing values in the diag1, diag2, and diag3 features with most common values (mode of the data).

Knowing the severity of the disease is important to identify a patient's health and should be proportional to the time spent in hospital, number of medications, and lab procedures. So, we create a new column 'Severity' which is the sum of time in hospital, number of procedures, medications, lab procedures, and diagnoses the patients are treated during the current admission. We have also created a new variable 'Visit Frequency', as the sum of the total emergencies, inpatient and outpatient numbers. If the patient frequently visits the hospital, he's likely to readmit quickly. Upon exploratory research found that patients who change medications tend to readmit more often. To capture this, we have engineered a new column 'Num Changes' to capture the changes (up/down) of medications for each patient.

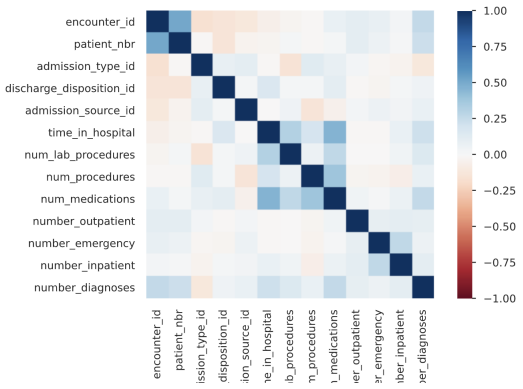


Figure 3: Pearson correlation of various columns.

3 Methods and Results

As shown in Table 1, there is an imbalance in the readmitted classes. To deal with unbalanced data we used different sampling techniques like oversampling, SMOTE. We then divided our dataset into training and test sets using a 80:20 split. Thus out of total 70431 observations, our training set contained 56344, and test set contained 14086 observations. We then trained different Classification models with 5 fold cross-validation.

Model	AUC(Test)	Accuracy	Precision	Recall
Naive Bayes	0.540	0.521	0.090	0.553
Logistic Regression	0.576	0.692	0.175	0.373
Neural Networks	0.658	0.801	0.264	0.430
Decision Trees	0.551	0.830	0.211	0.128
Random Forests	0.667	0.877	0.352	0.260
AdaBoost	0.611	0.617	0.181	0.519
XGBoost	0.615	0.627	0.180	0.502
CatBoost	0.623	0.621	0.182	0.545

Table 2: AUC-scores, accuracy, precision, recall for different methods, evaluated after balancing readmission rates

After following the above mentioned preprocessing steps, we converted all categorical variables into one-hot encodings. We experimented with Naive Bayes by assuming all the features are independent of one another and tried various versions of Naive Bayes and were able to only obtain an accuracy of 0.52, which also suggested that there was no direct correlation between readmissions and most of the columns. We also tested with Logistic Regression by using liblinear solver and trying out both L1 and L2 regularizations, with lambda ranging from 0 to 20. We achieved an accuracy of 0.69 which was better than Naive Bayes, but we observed that the precision and recall values of both the models was significantly low, which suggested that we should try more complex models. Then we trained Neural Networks varying network architecture upto 3 hidden layers and 30 nodes limit in each. We achieved maximum test AUC 0.658 with optimal parameters of 1 hidden layer with 15 nodes which yielded accuracy of 0.801.

For the next set of models, we used sklearn.model_selection.GridSearchCV to tune the best hyperparameters values to get better prediction results. First, we ran Decision Tree's, that work by recursive binary splitting at each node based on a test condition. We varied model parameters max_depth with range(1, 100, 1) and min_samples_split range(2, 20, 1) on our preprocessed train data. We found optimal parameters 'max_depth': 38, 'min_samples_split': 2 which yielded an accuracy of 0.83 for Readmitted values. As an extension of the Decision Trees next worked on the Random Forests algorithm, which is an ensemble method formed on bootstrapped samples from the dataset. For Random Forests we again conducted experiments varying max_depth in [5, 7, 10, 15, 20] and n_estimators range(100, 220, 30) with 5-fold cross-validation on train data. Figure 4 shows the observed AUC values from the experiments. We achieved an accuracy of X with optimal parameters as 'max_depth': 20, 'n_estimators': 190. We observed that the random forests model achieved the overall highest objective accuracy out of all the other models.

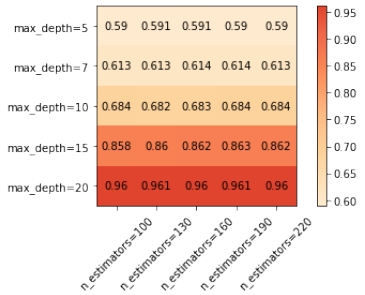
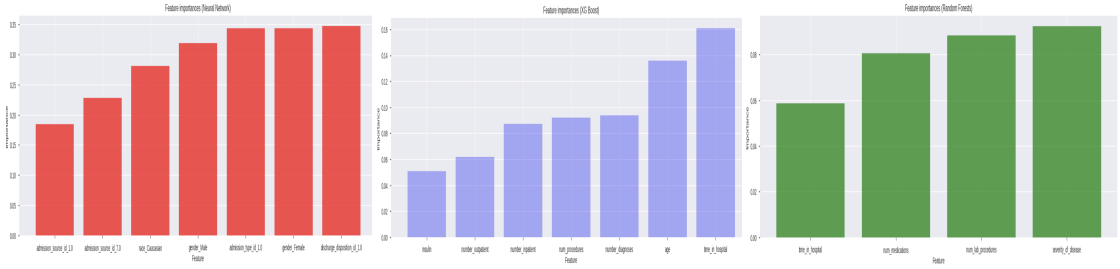


Figure 4: Hyperparameters Tuning using GridSearchCV AUC for Random Forests

Next, we attempted different boosting algorithms for classifying readmissions on our data. First, the AdaBoost (Adaptive Boosting) algorithm yielded an accuracy of 0.617 for Readmissions. But AdaBoost algorithm efficiency is highly affected by outliers as it tries to fit every data point perfectly with boosting. Next we tried XGBoost algorithm, an improvised version of the gradient boosting algorithm and achieved an accuracy of 0.627. We also ran experiments for the CatBoost algorithm varying iterations with range(100, 200, 15) and depth in [2, 4, 8] with 5-fold cross-validation on train data. We achieved an accuracy of 0.621 with optimal parameters as 'iterations': 130, 'depth': 4. We observed that the performance of XGBoost, AdaBoost, and CatBoost are very similar on data.

As mentioned above, we used a variety of different classification models to assess the prediction values. Based on our analysis of validation AUC, accuracy, and confusion matrix values, we determined that the best model was the Random Forests. Which yielded a good test accuracy of 0.877 on our dataset for Readmissions.



(a) Neural Networks (b) XGBoost (c) Random Forests

Figure 5: Feature importance for various methods

4 Discussion

We analyzed the Pearson correlation to understand the connection between various columns. From Figure 3, we can see that there is no significant correlation on readmissions with any standalone features but there are noticeable correlations among various columns, indicating scope for feature reduction. So, we analyzed the feature importance in various algorithms like Neural Networks, Random Forests, and Gradient boosting. From Figure 5, we see that from various algorithms, gender, race, total number of medications, lab procedures, admission type, time in hospital had a significant influence in predicting readmissions.

From the above experiments, we observed that although the accuracy values are fairly high, the precision and recall rates were not as expected. Although we used the sampling techniques to create synthetic data to compensate for the skewness in the data, it helped learn the decision boundaries better but could not improve precision and recall values. Possible future works include feature reduction, and improving models for precision.

5 References

[1] World Health Organization, Global report on diabetes. World Health Organization, 2016.

[2] N. Bhardwaj, B. Wodajo, A. Spano, S. Neal, and A. Coustasse, "The Impact of Big Data on Chronic Disease Management," Health Care Manag. (Frederick)., vol. 37, no. 1, pp. 90–98, 2018.

[3] Anika L. Hines, Ph.D., M.P.H., Marguerite L. Barrett, M.S., H. Joanna Jiang, Ph.D., and Claudia A.Steiner, M.D., M.P.H. "Conditions With the Largest Number of Adult Hospital Readmissions by Payer, 2011", April 2014, Healthcare Cost and Utilization Project, https://www.hcup-us.ahrq.gov/reports/statbriefs/sb172-Conditions-Readmissions-Payer.pdf

[4] Medicare.gov, "30-day unplanned readmission and death measures," 2017.

[5] H. Zhou, P. R. Della, P. Roberts, L. Goh, and S. S. Dhaliwal, "Utility of models to predict 28-day or 30-day unplanned hospital readmissions: an updated systematic review," BMJ Open, vol. 6, no. 6, p. e011060, Jun 2016.

6 Task splitting

Our work was collaborative and was split evenly. Details of the individual tasks:
Rajasekhar : Worked on the Naive Bayes, Logistic Regression and Neural Network models
Agniraj: Decision Trees, Random Forests, and boosting models(Adaboost, XGBoost, and CatBoost)
Shravan: Worked on ensemble models and visualizations
Data exploration, preprocessing and project writeup was a collaborative effort.