Predicting of Impact on Re-admission Rates for Patients Hospitalized with Diabetes

GANESH KASTURI

ganeshkasturi.datascience@gmail.com

INTRODUCTION:

During past decades, hospital readmissions have been the subject of retrospective surveys and prospective trials with a view to their prevention [1]. A hospital readmission is when a discharged patient gets re-admitted to a hospital within a certain period of time. The need for hospital readmission for certain conditions indicates the hospital quality. In other word, it shows inadequate care was provided to the patient at the time of first admission, and thus poses threat to patient life. Additionally, high hospital readmission rates affect the cost of care adversely. Particularly, early hospital readmissions, usually using the benchmark as less than 30 days, have been recognized as a common and costly occurrence among elderly and high-risk patients [2]. It reported that 20% of beneficiaries are readmitted with 30 days at a cost of more than $26 billion per year [3]. For diabetes patient particularly, American hospitals spent over $41 billion on diabetic patients in 2011 who got readmitted within 30 days of discharge [4]. Therefore, to improve the quality of care and a reduction in unnecessary health expense, United States congress passed the Hospital Readmission Reduction Program (HRRP). Starting in October 2012, the Centres for Medicare and Medicaid Services (CMS) began financially penalizing hospitals that perform worse than the national average on risk standardized readmission rates for Medicare patients, using 30-day as a threshold [5]. This lends the critical value for understanding the readmission rate within 30 days, and among our interest group, diabetes patients. Therefore, identifying patients at high risk early in hospitalization will help to reduce the readmission rate, in that hospitals can focus on preparing readmission for patients at high risk to shorten the length of readmission.

Addressing this critical issue involves the intensive data analysis throughout the research process. This study is a secondary analysis using machine learning methods. Our goal of the analysis is to find the determining factors that lead to higher readmission and correspondingly being able to predict which patients will get readmitted. Therefore, we proposed two research questions:

- What methods can we use to best predict hospital readmission in this dataset?
- What are the strongest predictors of hospital readmission in diabetic patients?

The rest of the paper is organized as follows: Section 2 will present a brief overview of the past work and identify the research gaps. Section 3 will specify the methodology we used in the study, including the dataset description and the analysis process. Such process includes data processing, explorative analysis, feature engineering, and modelling and evaluation.

METHODOLOGY:

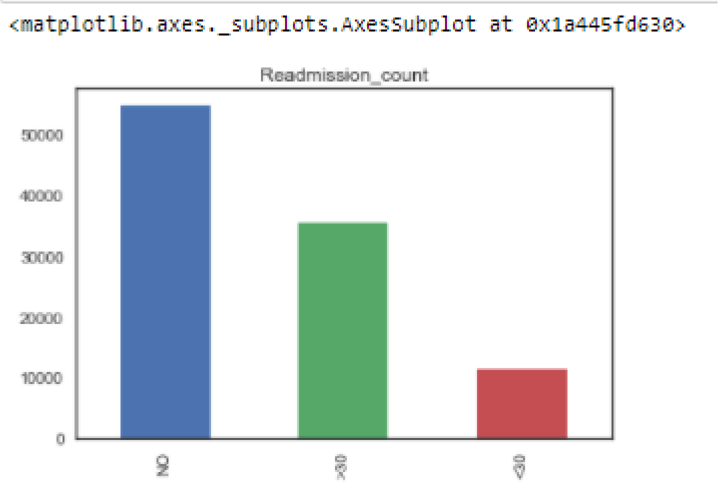
In this section, we will provide a description of the dataset, the exploratory data analysis, feature engineering, modeling and evaluation.

Data Set

To explore this problem, we used a secondary dataset from UCI machine learning repository [14] dataset. The dataset includes 101,766 instances, representing 10 years (1999- 2008) of clinical care at 130 US hospitals and integrated delivery networks across the Midwest (18 hospitals), Northeast (58), South (28), and West (16). Most of the hospitals (78) have bed size between 100 and 499, 38 hospitals have bed size less than 100, and bed size of 14 hospitals is greater than 500. The features collected in the dataset are related to patient's demographic information such as race, gender, age, weight; the information related to their hospital diagnosis and treatment, such as num_lab_procedures, num_medications, num_outpatient, diagnosis and medication prescription. The dataset is just an extracted subset set of Health fact dataset. Given this is an open dataset that include the longitudinal and cross-sectional data, and with relatively complete attributes (55 attributes), and released in the recent year (2014), we chose the dataset for exploring the questions.

Exploratory Analysis

Prior to performing any analysis, we conducted exploratory analysis to preview the data type, attributes, and overall patterns of the data. We are interested in the class label “Readmitted” (See Fig 1), so we checked the distribution of the readmitted, and several categorical variables. For the numerical variables, we used scatter plot to demonstrate the relationship between numerical variables and their distributions



```
<matplotlib.axes._subplots.AxesSubplot at 0x1a445fd630>
```

Figure 1: Bar plot for the class label.

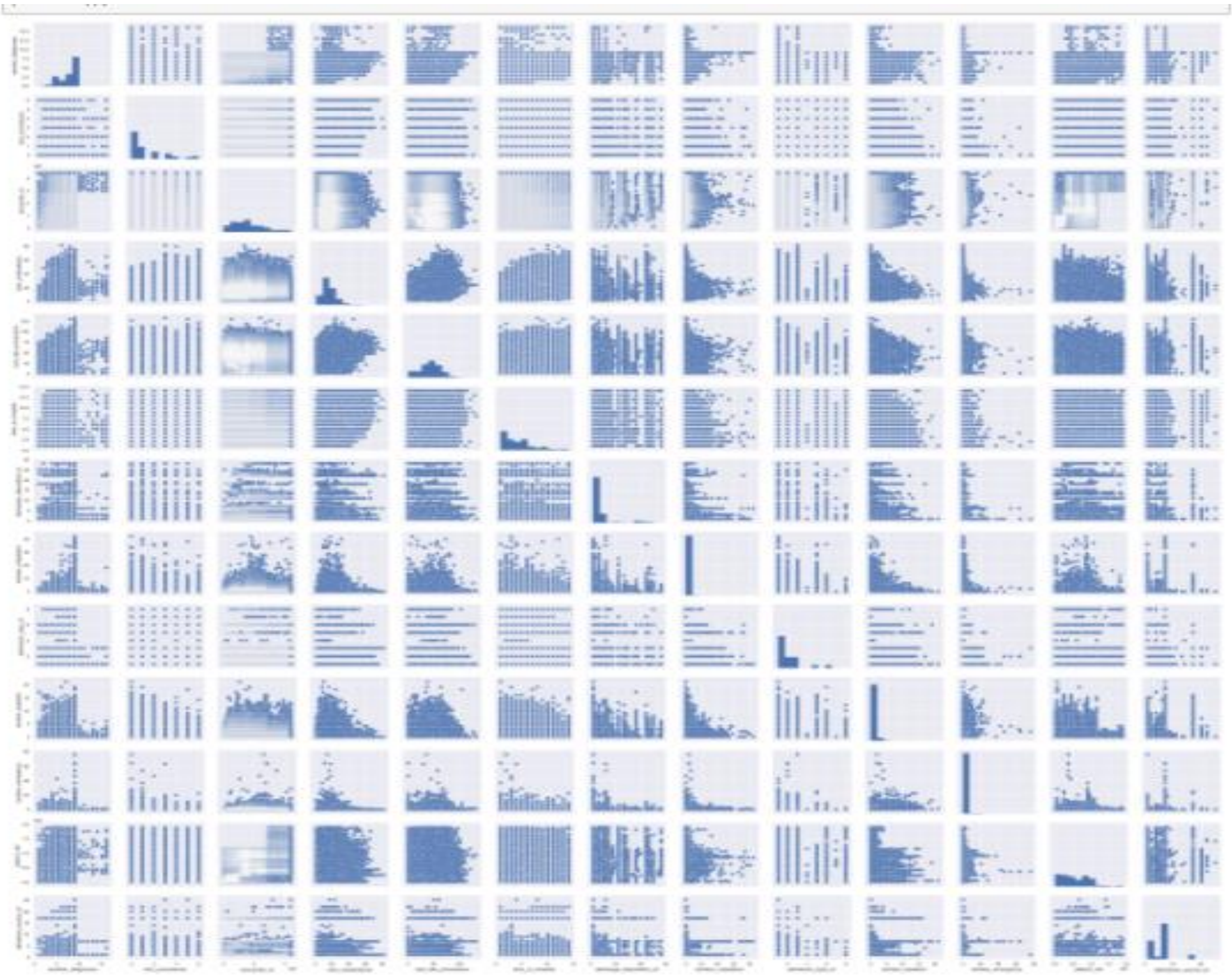


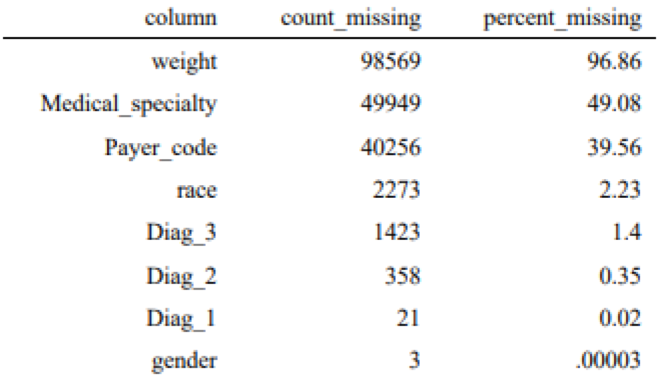
Figure 2: Scatter plot for numeric features.

Data Pre-Processing

After the exploratory analysis, we found several challenges lies in the original dataset, and thus some data wrangling tasks such as data cleaning, dealing with missing values, creating new variables, and data transformation needs to be addressed before modelling. The tools used here are Python Packages including NumPy, Pandas, Matplotlib and Seaborn. Specifically, we conducted several pre-data processing processes:

Dealing with Missing Data

We discovered a large number of missing values coded as "?" across nominal variables. As Table 1 shows, this dataset has 8 variables which contain missing values. Since weight, medical specialty, and payer code contains over 35% values, and also because of the irrelevancy toward our study, we decide to drop all of them. Race only includes 2.23% missing values, so we only drop the missing values and keep the rest. Primary (diag_1), secondary (diag_2) and additional (diag_3) diagnoses each has less than 2% missing values, but compared to the total number of instances, we still need to clean them. Technically, our goal is to maintain the most information of the dataset, especially the diagnosis is an important variable related to the diabetes patients. Therefore, we adopted a strategy to drop the missing values when all three diagnosis were missing. We then only drop 3 unknown and invalid instances in our dataset.



column	count_missing	percent_missing
weight	98569	96.86
Medical_specialty	49949	49.08
Payer_code	40256	39.56
race	2273	2.23
Diag_3	1423	1.4
Diag_2	358	0.35
Diag_1	21	0.02
gender	3	.00003

Dropping Attributes

After a quick view of the current dataset, we found some patients died during the hospital admission who do not have any probability of being readmitted, so we removed those tuples, as the discharge_disposition_id=11. We also drop two variables (drugs named citoglipton and examide) in which all records have the exactly same value. By noticing that two variables called encounter_id and patient_nbr has no relevance with the class label readmission, so we also drop those two variables.

Creation of New Features

1) patient_service: We created a new feature called patient_service, which measures the total number of hospital/clinician services a patient used in the past year. This feature is the sum of original variables for number of inpatient visits, emergency room visits, and outpatient visits. We did not apply weighting for these three variables. The reason for the creation is to lower the dimension of our data and try to make the dataset simpler.

2) med_change: The dataset contains 23 medications of the medicine use for a patent during the stay in hospital. Each of the tuple records when a change was made in this medication or not during the current stay as No-for no medication, up for increasing the dose, Down-for decreasing the dose and Steady-for keeping the current dose. Instead of counting changes for each medication, we decide to combine them and count changes for all of them. We define No and Steady as no change, while up and down for change. Doing this step will simplify the model and we can try to find out if the readmission is related with medication changes.

3) num_med: Not only medication changes can be related with readmission, the total number of medications used can also be a key feature, due to the fact that the number of the medicine reflected the severity of certain disease. And thus, we created a variable called num_med in order to store the total number of medications a patient used during the stay of hospital.

Feature Engineering

Data Type Conversion

For nominal features, we converted them into object type, in order for the later numerical variables processing

Log Transformation, Standardization, and Correlation

For numeric features, the scatter plot of the distributions as Figure 1 indicated most of numerical are highly skewed and had high kurtosis. Using the threshold=+-1 as skewness for normal distribution, if skewness is less than -1 or greater than 1, the distribution is highly skewed. If skewness is between -1 and -0.5 or between 0.5 and 1, the distribution is moderately skewed. If skewness is between -0.5 and 0.5, the distribution is approximately symmetric. As the standard for kurtosis, the threshold = 3 is for normal distribution. Therefore, we used log transformation to normalize the numerical variables to make sure numeric variables had a Gaussian-like or normal distribution.

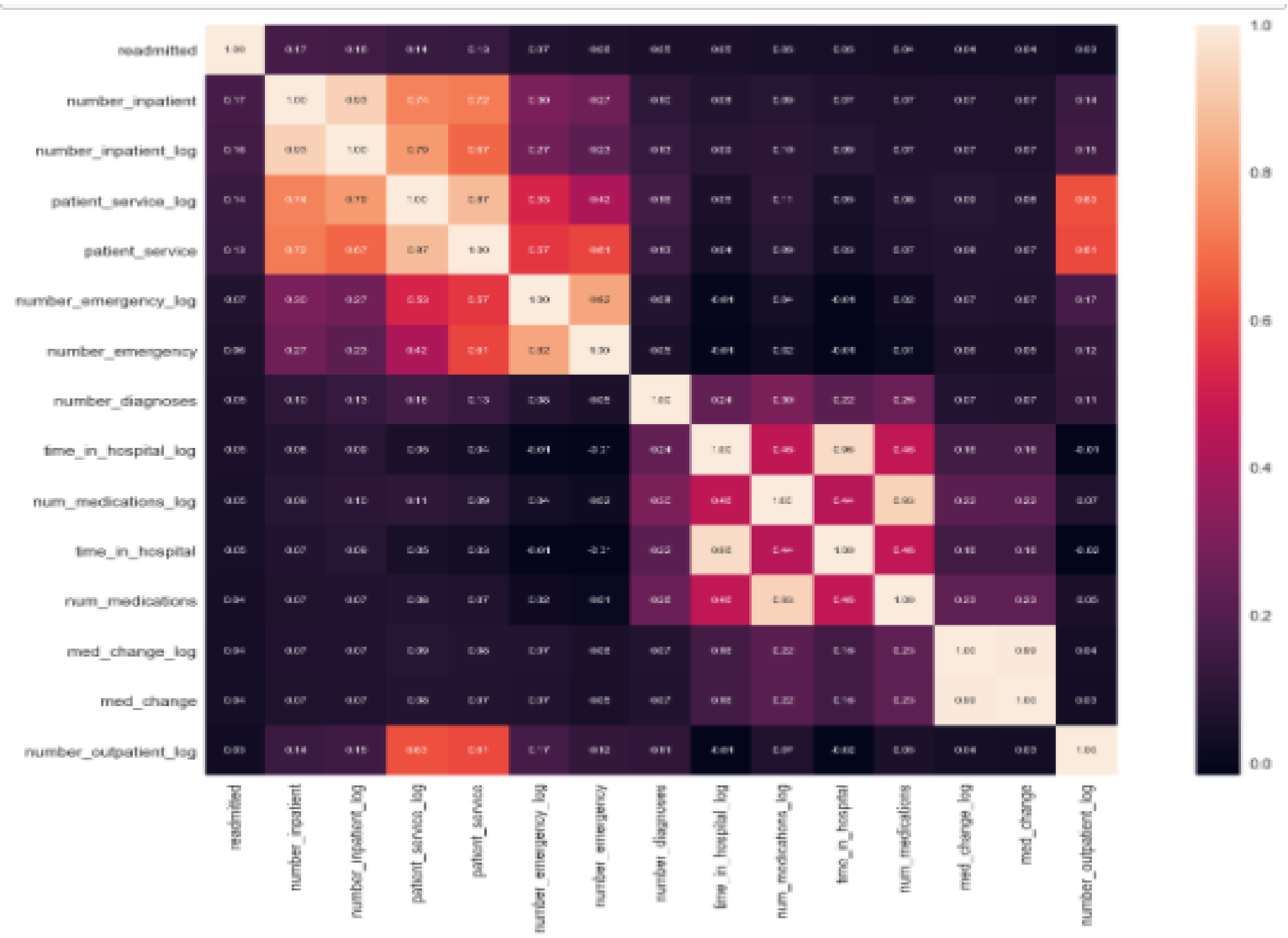
Since the numerical variable are not using the same scale so we rescale our data using the standardization methods with the following formula:

$$New\ value = \frac{Value\ -\ Mean(Values)}{Standard\ Deviation\ (Values)}$$

After all data are standardized, we checked the correlation between the variables using a heat map to find top 15 correlated variables as Fig. 3 shows. There is not too much correlation between the variables and the correlation listed are self-explainable.

Outliers.

For detecting and processing the outliers, we used the coverage rule for normal distribution to deal with outliers. As Fig. 4 shows, the remaining 0.3% of the data are treated as outliers for this project. And thus, we removed the outliers.



Class Imbalance

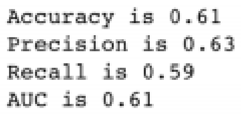
Before modelling, we checked class label balance after running analysis to check the class balance. As a result, we see there are 79512 tuples belong to class 0, which is >30- day readmission and no admission needed. Only 9607 patients belong to. We used the confusion matrix to ensure the class label is highly balanced as Fig. 5 shows. Our benchmark model logistic regression showed that 89% accuracy with precision and recall rate equals to zero. We used an oversampling technique (SMOTE) to our data is balanced by oversampling our underrepresented class of readmissions. Fig. 6 explains how over-sampling and under-sampling works. After using SMOTE, we will get 79512 patients with category 0, and 79512 patients with category 1 too. Fig. 5 also shows the confusion matrix before and after data balancing.

EXPERIMENT

Our aim for experiment of modelling is to identify the factors for high-risk diabetic patients, which was posed as the problem of classifying of whether a patient would be readmitted within 30 days of being discharge or after 30 days of being discharged or never readmitted. Therefore, different classification algorithms are adopted for experimentation and find the best method that achieve the highest accuracy. We adopted and compared the four classification algorithms. Prior to training the classification algorithms, we randomly split our dataset into two distinct sets - the training and the test set. The training and test set consisted of 90% and 10% of the data accordingly. The parameters of each algorithm were chosen based on the classification performance evaluated by 10-fold-cross-validation on the training set. The performance of all algorithms was evaluated on the test set. The methods we implemented include:

Logistic Regression

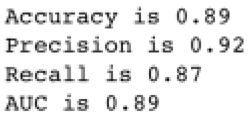
Logistic regression is used as a benchmark model for our analysis. Since we assume that our data can be modelled as a log likelihood of outcome for the binary class label readmission, logistic regression can help us to understand the relative impact and significance of each attributes. We test this model by using 90% training and 10% testing data and 10-fold cross-validation. We achieved a cross validation score: 61.29% and test set score 61.35%. By looking into the confusion matrix, we can calculate several measures of accuracy:



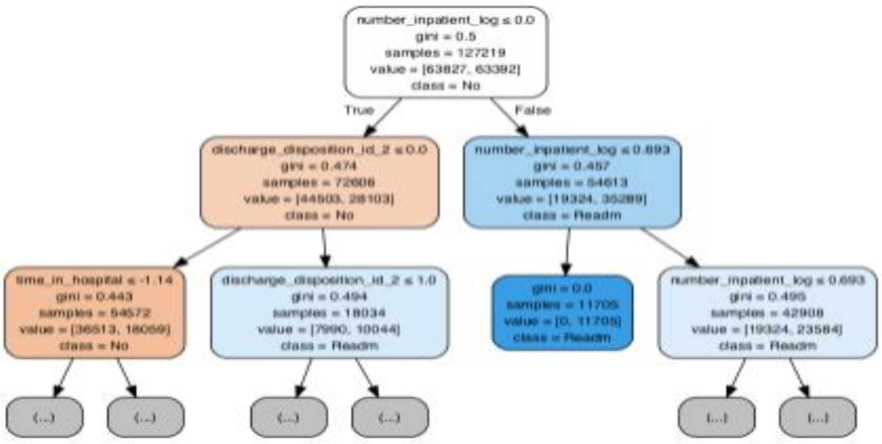
```
Accuracy is 0.61
Precision is 0.63
Recall is 0.59
AUC is 0.61
```

Decision Trees

Decision trees is a popular tree-based model that is easily to interpret the logic for splitting. Decision trees classify the data by sorting them down the tree from the root to some leaf node, with the leaf node providing the classification to the data. Due to the interactions between variables inherently, we removed the interaction variables from the feature set we did for logistic regression. Similarly, we did 10- fold cross validation score for decision trees too. The score equals to 88.97% and the dev set score is 89.43%, so decision trees look good for this dataset. After checking the score, we analysed the confusion matrix for decision trees for both entropy and Gini methods. As a result, both of them yielded the same results of measurements:

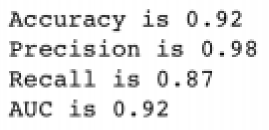


```
Accuracy is 0.89
Precision is 0.92
Recall is 0.87
AUC is 0.89
```



Random Forests

Random Forest is composed of a set of decision trees. Each decision tree acts as a weak classifier and pooling the responses from multiple decision trees leads to a strong classifier. Each decision tree is trained independently and determines the class of an input by evaluating a series of greedily learned binary questions. The random forest consisting of 10 trees, with the max_depth of as 25 nodes was used, as it was found to be optimal from the experiment with varying number of trees and depth in the forest. After implementing Random Forest, we achieved similar results of measurements for using Gini and entropy methods. Random Forest showed better results than decision tree as regards to prediction accuracy.



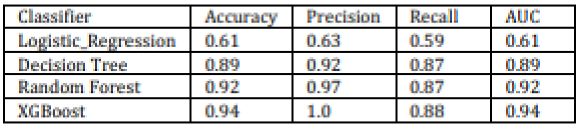
```
Accuracy is 0.92
Precision is 0.98
Recall is 0.87
AUC is 0.92
```

Model Improvement

After running the random forest, we decided to use a boosting method by the relatively new algorithm XGboost for model improvement. Boosting is an ensemble method that create a strong classifier based on weak classifiers, according to how correlated are the learners to the actual target variable. The errors of the previous model are corrected by the next predictor, by adding models on top of each other iteratively until the training data is accurately predicted or a maximum number of models are added.

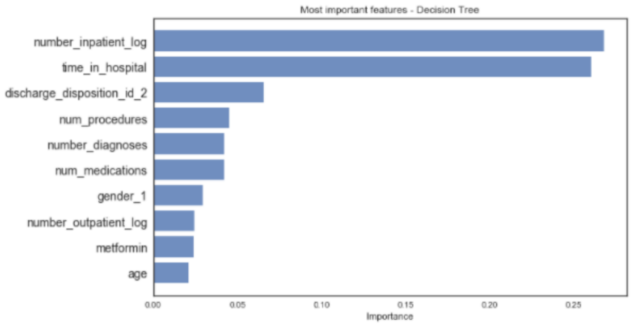
Classifier Comparison

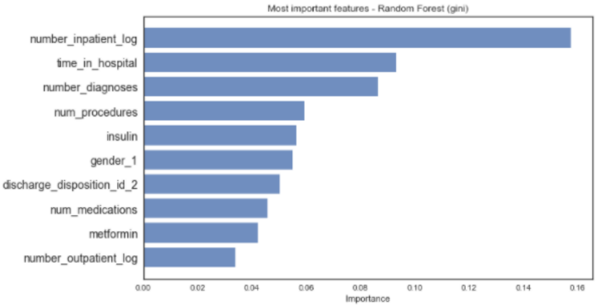
Each algorithm was evaluated using a 10-fold stratified cross-validation. Cross-validation is an evaluation technique where the dataset is randomly but evenly distributed into a number of folds. Stratified cross-validation attempts to preserve the class distribution between folds so that each fold is representative of the full dataset. The learning algorithm is trained on 9-fold and tested on 1-fold of dataset. Repeating the cross-validation process ensure that particular random initialization does not bias the overall result

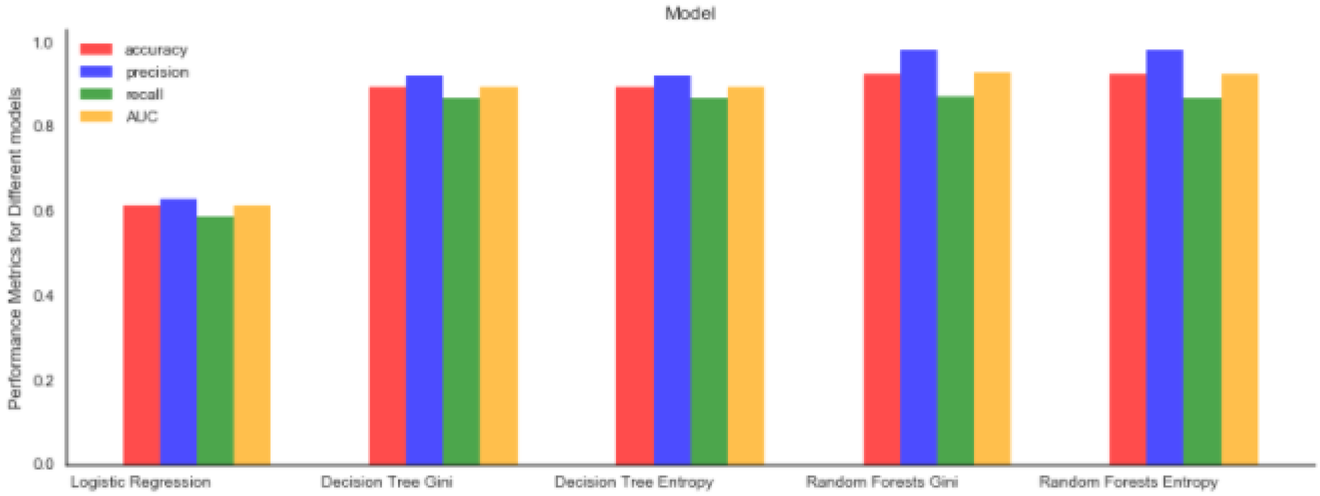


Classifier	Accuracy	Precision	Recall	AUC
Logistic_Regression	0.61	0.63	0.59	0.61
Decision Tree	0.89	0.92	0.87	0.89
Random Forest	0.92	0.97	0.87	0.92
XGBoost	0.94	1.0	0.88	0.94

Most Important Predictors







CONCLUSION

In this work we adopted machine learning methods to identify high risk patients and evaluated different machine learning algorithms. Compared to the previous analysis, our study achieved high accuracy due to the sophisticated pre-processing procedure. The XGBoost method is reported to be the best method for prediction of the readmission rate for diabetes patients. We identified the most important factors as the time_in_hospital and number of inpatients, number of diagnosis, which appears to associate with the severity of the disease. Further studies could conduct more exploration when analysing these factors individually.