Diabetes Patients Early Readmission Prediction

Peter Mačinec and František Šefčík

Faculty of Informatics and Information Technologies,
Slovak University of Technology, Bratislava

Abstract. Nowadays, more and more patients suffer from still incurable diabetes disease. Every wrong chosen treatment for patients can harm their health and lead to early readmission that costs more money. Therefore, there is a demand for predicting the readmission of patients to increase quality of health care and also to reduce costs. However, standard methods for identifying the patients with risk of readmission perform poorly (e.g. LACE index). With growing number of patients with diabetes, there is a need for methods that can automatically and more accurate predict the readmission. In this paper, we provide method based on machine learning for predicting early readmission of patient. The results of data analysis already showed that there is a potential of using data-driven approach for this problem.

Keywords: diabetes · early readmission prediction · machine learning · data analysis.

1 Motivation

Diabetes is a wide spread chronic disease that is related to irregular blood glucose levels caused by problems with insulin. The number of people with diabetes has increased enormously in recent years and costs for health care with each hospital admission are rising simultaneously. The methods of diabetes treatment of patient have high impact on mortality and morbidity. Wrong treatment can endanger patient's health and may lead to early readmission.

A hospital readmission is when a patient who is discharged from the hospital, gets re-admitted again within certain period of time. By predicting readmission, more attention may be given to treatment of patients with high probability of readmission and so increase the quality of care during hospitalization. Because there is no cure for the diabetes yet[1] and diabetic patients can be readmitted in the future, an *early readmission prediction* can help mostly when it comes to selection of best treatment for the patient.

Data of patients clinical encounters are being collected naturally with healthcare systems, thus data-driven approach seem to be appropriate for this problem. Machine learning algorithms used for early readmission prediction provide ability to process the data of a lot of patients and may help to find hidden dependencies in the data to outperform basic methods (e.g. LACE index).

[1] https://www.diabetes.org.uk/diabetes-the-basics/is-there-a-cure

The task of *early readmission prediction* can be represented as binary classification problem into two classes - patient was early readmitted or not. The term *early readmission* is very relative. In our case, we define it as patient being readmitted in less than one month (30 days).

2 Related works

Task of predicting diabetic patients readmission is substantive from two points of view - health of patients and saving money because of readmission. Because data of patients are usually available from health records, much research has been done also in data mining area.

Majority of previous works in this area are based on traditional machine learning workflow, including data analysis, data preprocessing and modeling. However, the task defined can vary across researchers - some of them are trying to predict early patients readmission [2, 4, 3], and then there is also work where authors are trying to predict short and long term readmissions [1]. Different but interesting approach that was also the subject of research is to predict readission across age groups [3].

In the previous works, much effort has been spent on data preprocessing. Basic well-known steps of preprocessing like filtering useless columns, normalization, outliers removal and one-hot encoding were usually performed [4, 3, 2, 1]. Also heavy ensemble models have been tried instead of detailed preprocessing [3], but comparison of the achieved results with other works showed that the proper preprocessing always has its place in machine learning tasks.

Because the classes (patient was early readmitted or not) are naturally imbalanced, class balancing was usually performed. Oversampling was preferred over undersampling, and usually SMOTE (Synthetic Minority Over-sampling Technique) was used [4, 2]. From the analyzed works we can deduce that training machine learning algorithms after data oversampling leads to better results.

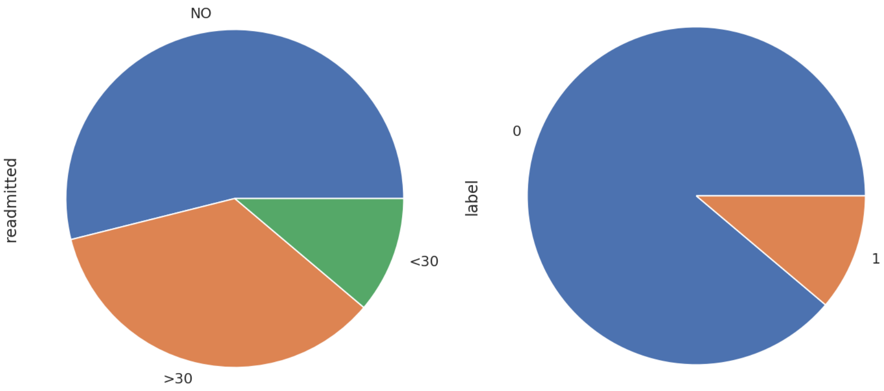
When it comes to modeling, many different approaches and methods were used. Mostly used were tree-based algorithms, like random forest [4, 1], decision tree [4] or xgboost [4]. Also, other algorithms were tried, e.g. logistic regression [4], Adaboost [4, 1], naive Bayes [1]. Neural networks are not so popular in this area, but for example multi-layer perceptron [1] or convolutional neural networks [2] were used. In general, tree-based models (and mostly random forest) alongside with convolutional neural networks achieved the best results - accuracy about 90%-94% after data balancing.

3 Dataset

To evaluate our method of diabetes patients early readmission prediction, we have chosen real world dataset of diabetes patients clinical encounters [5]. The dataset was created from Health Facts database with data collected during 10 years (1999-2008) across 130 hospitals in United States. From this large-scale database with millions of records, final dataset with 101 766 records of patients encounters was derived using 5 criteria defined by authors (e.g. it is diabetic and inpatient encounter, or that laboratory tests were performed during the encounter, etc.).

The dataset contains 50 attributes (features), both numerical and categorical. Attributes are of various types - demographics of the patient (like *race*, *gender*, etc.), diagnoses, diabetic medications or number of visits in the preceding year. All of these attributes have been chosen by clinical experts to be potentially associated with patient's diabetic condition.

As shown on Fig. 1a, 3 classes describing whether and when was patient readmitted are provided for prediction. According to the authors of the dataset [5] and our task definition, we transformed the problem to binary classification of *early readmission*. In this scenario, records of patients readmitted in less than 30 days are considered to be in positive class (early readmitted), othervise not early readmitted. Final distribution of the classes after task adjustment is shown on Fig. 1b. As we can see from the figure, classes are highly imbalanced. We provide also detailed analysis of the dataset and its attributes[2].



(a) Original classes distribution

(b) Binary-converted classes distribution

Fig. 1: Distribution of predicted classes - original (whether and when was patient readmitted) and after converting to binary form (only whether was patient readmitted or not).

4 Data preprocessing

During analysis of our dataset, some problems with data have been identified. To ensure reproducibility of preprocessing, data were preprocessed via *pipeline*. Basic preprocessing steps based on data analysis results are included in pipeline:

[2] https://github.com/pmacinec/diabetes-patients-readmissions-prediction

1. Drop redundant columns (or those with too low diversity, e.g. containing one major value). The columns with more than 45% of missing values are dropped and columns where major value is present more than 90%. Also columns *encounter id, patient nbr, payer code* that we observed as redundant are dropped.
2. Merge too small classes in categorical attributes into one *other* value, if there is less than 5% values for class.
3. Fill missing values (most-frequent value for categorical attributes and median for numerical ones),
4. Feature engineering (new features identified in data analysis). From attributes *number emergency, number outpatient, number inpatient* and all attributes describing medicament we created new *visits sum, number medicaments changes, number medicaments* attributes. Also we map diagnoses codes to diagnoses categories according to data analysis.
5. Map ordinal features values into numbers (age categories we map into number values ordinary),
6. One-hot encoding (nominal attributes into numbers).

We have checked numerical attributes carefully to see whether normalization and outliers removal should be performed. According to analysis, there are no such differences in measures so normalization is not needed. We have checked also outliers, but we have not found any extreme value that should be removed.

Some new features have been explored - total number of visits, number of medicaments changes and total number of medicaments used. Also, diagnoses codes had to be mapped into diagnoses categories (according to original paper [5]).

5 Methods

Our method to help to solve the task of early readmission prediction is based on basic machine learning workflow, using variety aspects of data analysis and machine learning.

With properly preprocessed data we did first experiments with three different types of machine learning models like Random forest (RF), XGBoost and Multilayer perceptron (MLP).

Random forest belongs to the group of tree based algorithms, which are one of the most popular in solving all kinds of data science problems. The building block of RF is Decision tree. Decision tree is tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a class label.

A tree is learned by splitting training set into subsets by set of splitting rules based on classification features. This process repeats recursively on each derived subset while node contains only samples from one class or when splitting no longer adds value to the predictions.

RF is ensemble of many Decision trees. When RF predict class label, result is obtained by voting of all trees in the ensemble. We decided to use RF because performs very well on the very large volume of data with high dimensionality. Next benefit is extraction of feature importance which can be helpful in models improving and feature selection.

XGBoost is also tree based algorithm containing set of Decision trees as above mentioned Random forest. These models are very similar but main differences are in a way how trees are built and combining results.

Random forests builds each tree independently while gradient boosting builds one tree at a time, where each new tree helps to correct errors made by previously trained tree. Random forests combine results at the end of the process while gradient boosting combines results along the way.

We decided for XGBoost because it is an implementation of gradient boosted decision trees designed for speed and performance. Also it can has better performance if parameters are carefully tuned and it performs well on unbalanced data what is our case. Prone to over-fitting is one of the disadvantages of gradient boosted algorithms.

MLP is a class of feedforward artificial neural network, containing at least three layers. One input and output layer with multiple hidden layers, where each layer consist multiple neurons. Neurons from one layers are fully connected with neurons from previous and next layer. Each neuron is weighted sum of outputs from neurons in previous layer with linear activation function which process output of neuron.

Learning of MLP is based on changing connection weights in every data iteration, based on the amount of error in the output compared to the expected result. This is called loss, which is backpropagated and the weights of the model are updated by using gradient.

For reason we identify neural networks in related works as one of the solution with highest precision, we chose MLP as promising solution to problem of early readmission classification.

6 Evaluation

For the evaluation phase we focused on early defined experiments with three types of machine learning algorithm. In the first step we split data into train and test set in the ratio 80%-20%. Because the data are highly unbalanced, where the 89% of data are from *late readmission* class and 11% from *early readmission* class, we decided to apply undersampling and oversampling methods for experiments.

As undersampling technique we used *NearMiss* that select examples based on the distance of majority class examples to minority class examples. There are three versions of this algorithm, we decided for NearMiss-3 version, which select a given number of majority class examples for each example in the minority class that are closest. In case of oversampling we chose SMOTE technique, which create new samples based on feature space selecting. SMOTE draws the line between tho closest examples in feature space and generate a new sample at a point along that line. With these data balancing techniques was balanced only training set and test set stayed unbalanced.

To complex comparison of how model is good, we used several evaluation metrics like Accuracy, F1 score, Precision, Recall and AUC score. We selected these metrics because we need get better look how model perform on each class. Accuracy is not suitable for our problem, because of highly imbalanced data. Precision and Recall give us more realistic look at the model but only on one predicted group. However, AUC ROC appears as the most accurate metric to finally compare models on unbalanced dataset and our classification problem. AUC ROC is metric that is used to measure how well the model can distinguish two classes.

In the Table 1, you can see results of first experiments with Random forest, XGboost a MLP models. For each model we were manually setting a hyper-parameters to achieve the best performance. Also we experimented how model perform with original, undersampled or oversampled data. We achieved best results with Random forest on original data, where we manually tuned parameters and most important step was to set class weight parameter to balance learning ratio for each class. Very similar results achieved XGBoost with original data, where we also needed to set parameters to balance weight for classes. With MLP we couldn't achieve better results as before mentioned models, what we attribute to unbalanced data and no option to set some parameter of model which would by discriminate class unbalancing.

For second experiment we were observing data balancing techniques and their addition to model performance. From Table 1 we can read that no of models with data balancing technique achieved better results as with original data.

model	Accuracy	F1 (micro)	F1 (macro)	Precission	Recall	AUC ROC
RF	0.60	0.60	0.49	0.16	0.62	0.61
RF under	0.37	0.37	0.33	0.10	0.56	0.46
RF over	0.47	0.47	0.41	0.13	0.68	0.56
XGBoost	0.48	0.48	0.43	0.14	0.74	0.60
XGBoost under	0.38	0.38	0.34	0.10	0.55	0.46
XGBoost over	0.54	0.54	0.45	0.14	0.58	0.56
MLP	0.89	0.89	0.48	0.37	0.01	0.50
MLP under	0.47	0.47	0.39	0.10	0.50	0.48
MLP over	0.89	0.89	0.48	0.20	0.01	0.50

Table 1: Results of experiments. (under- undersampled data, over-oversampled data)

7 Conclusion

In our work we did comprehensive analysis of diabetes patients clinical encounters dataset [5] with more than 50 features. Based on the analysis we properly preprocessed data, where we drop redundant columns, fill missing values, create new features, map data formats and apply one hot encoding. With prepared data we did preliminary experiments with three models - Random forest, XGBoost and MLP, trained also on original, undersampled and oversampled data.

In preliminary experimnets we achieved the best results with Random forest, with AUC score 0.61. XGBoost achieved very similar performance, which can be caused by fact that both models are tree-based. Slightly different results may be achieved if hyperparameter tuning would be performed for each model, but we do not expect significant differences.

Next observations were done on data balancing techniques NearMiss and SMOTE. From related papers the best results was achieved after using of SMOTE technique. In our case these techniques did not help to achieve better performance of models, what can by caused by samples being randomly placed in feature space. That means we can not clearly divide classes by any machine learning model. We analyzed several of related papers where data balancing techniques boost performance of model and we found out that all of those works have applied SMOTE on whole dataset before train-test splitting. This approach leads model to learn to classify SMOTE algorithm but not real classification problem.

The preliminary results show that no data-balancing technique can help to improve predictions. Future work will be rather focused on feature selection and hyper-parameter tuning.

References

1. Bhuvan, M.S., Kumar, A., Zafar, A., Kishore, V.: Identifying diabetic patients with high risk of readmission. ArXiv **abs/1602.04257** (2016)
2. Hammoudeh, A., Al-Naymat, G., Ghannam, I., Obied, N.: Predicting hospital readmission among diabetics using deep learning. Procedia Computer Science **141**, 484 – 489 (2018). https://doi.org/10.1016/j.procs.2018.10.138, the 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2018) / The 8th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2018) / Affiliated Workshops
3. Mingle, D.: Predicting diabetic readmission rates: Moving beyond hba1c. Current Trends in Biomedical Engineering & Biosciences **7** (01 2017). https://doi.org/10.19080/CTBEB.2017.07.555715
4. Sharma, A., Agrawal, P., Madaan, V., Goyal, S.: Prediction on diabetes patient's hospital readmission rates. In: Proceedings of the Third International Conference on Advanced Informatics for Computing Research. ICAICR '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3339311.3339349
5. Strack, B., Deshazo, J., Gennings, C., Olmo Ortiz, J.L., Ventura, S., Cios, K., Clore, J.: Impact of hba1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. BioMed research international **2014** (04 2014). https://doi.org/10.1155/2014/781670