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 and machine learning methods 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.

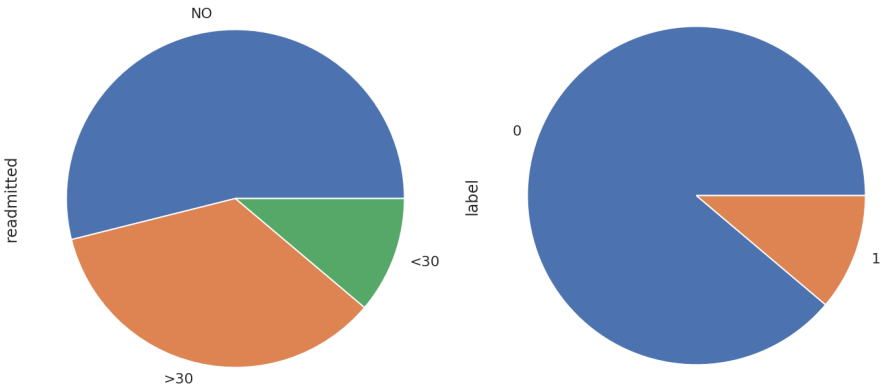
3 Dataset

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), otherwise 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 early 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 and columns where major value is present in more than 90% cases are dropped. Also columns *encounter id, patient number, 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 with most-frequent value for categorical attributes and median for numerical ones,
4. Feature engineering (adding new features identified in data analysis). From attributes *number emergency, number outpatient, number inpatient* and all attributes describing medicament we created new attributes: *visits sum, 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. 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.

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 mining methods.

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 in combining results.

Random forest 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 to use XGBoost because it is an implementation of gradient boosted decision trees designed for speed and performance. Also, it is known as very powerful algorithm if it's parameters are carefully tuned. However, 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, and multiple hidden layers consisting of multiple neurons. Each neuron in a layer is fully connected with all neurons from previous and next layer. Each neuron is weighted sum of outputs from neurons in previous layer with linear activation function.

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 have identified neural networks in related works as one of the most powerful methods, we have chosen MLP as promising solution to problem of early readmission prediction.

6 Evaluation

6.1 Methodology

In the evaluation phase, we mainly focused on previously defined experiments with three types of machine learning algorithms. In the first step we split the data into train and test sets in ratio 80%-20%. Because the data we have chosen are highly imbalanced and only 11% of the samples represent patient being early readmitted, we decided to experiment also with class balancing methods.

As undersampling technique we used *Random undersampling* that selects random samples from majority class to equal number of minority class samples. In case of oversampling we chose SMOTE technique that was mostly used in related works. SMOTE draws the line between the closest examples in feature space and generates new sample at a point along that line. With these data balancing techniques, only training set was balanced.

For complex comparison of how is each model performing, we used several evaluation metrics like F1-score, Precision, Recall and AUC ROC score. ROC curve consists of true positive rate (TPR or Recall, Sensitivity) and false positive rate (FPR or Fall-out) while positive class prediction probability threshold is varied. Figure 2 shows confusion matrix interpreting true positives, false positives, true negatives and false negatives.

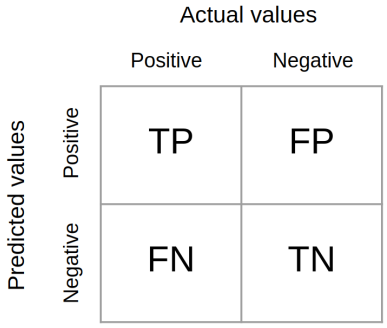


Fig. 2: Confusion matrix.

Then, TPR and FPR are defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

All of mentioned metrics were calculated considering each class individually except AUC ROC score (that was calculated only considering positive class). We omitted Accuracy score because of highly imbalanced data. Precision, Recall and F1-score offer more realistic look at the model performance on each predicted group, however AUC ROC appears to be the most suitable metric considering mainly (but not only) positive class. Considering that, we have chosen AUC ROC as the main metric for final models comparison.

6.2 Experiments results

Table 1 shows the results of first experiments with Random forest, XGboost a MLP models. For each model, hyperparameters have been tuned only manually by intuition in this phase. Each of the models has been trained on three types of the data - original, undersampled and oversampled (with techniques mentioned in 6.1). We have achieved best results with Random forest on original data (with class weight parameter set to *balanced*) and also the same model trained on undersampled data. Very similar results achieved XGBoost trained on original and undersampled data, alongside with MLP trained on all data variants. Results also showed that oversampling using SMOTE is not so much suitable for this data, even though majority of related works are using it. So, based on the results, we assume that the high scores achieved in the related works with SMOTE being used are due to oversampling also test set. We have tried also this setup and were able to achieve so high scores, but we consider this approach as incorrect.

Table 1: Results of first experiments - comparing Random forest, XGBoost and Multi-layer perceptron.
Note: All metrics are calculated when considering early readmitted patients as positive class.

Model	**F1 (micro)**	**F1 (macro)**	**Precision**	**Recall**	**AUC ROC**
RF	0.67	0.52	0.17	0.53	**0.65**
RF undersampled	0.60	0.49	0.16	0.63	**0.65**
RF oversampled	0.42	0.38	0.13	0.75	0.60
XGBoost	0.52	0.45	0.15	0.69	0.64
XGBoost undersampled	0.58	0.48	0.16	0.63	0.64
XGBoost oversampled	0.42	0.38	0.12	0.69	0.57
MLP	0.89	0.49	0.41	0.02	0.64
MLP undersampled	0.63	0.50	0.16	0.56	0.64
MLP oversampled	0.89	0.49	0.35	0.02	0.64

In next phase of experiments, we have tried also three ensemble techniques - *Bagging*, *Voting* and *Stacking*. We expected that any kind of several models combination can give as better and more stable predictions. In all of these ensemble-based experiments, algorithms were trained on undersampled data only, as this setup seemed to be the best (according to previous results).

As a base classifier for Bagging method, we selected Random forest with manually chosen hyper-parameters setup from previous experiments. For other both methods, Voting and Stacking, we used all previous tested models - Random Forest, XGBoost and MLP. In Stacking method, simple *Logistic Regression* was used to combine the classifiers. Table 2 shows results for all ensemble methods. As can be seen in the table, none of the ensemble methods significantly improved the prediction performance.

Table 2: Results of experiments with ensemble methods.
Note: All metrics are calculated when considering early readmitted patients as positive class.

Model	**F1 (micro)**	**F1 (macro)**	**Precision**	**Recall**	**AUC ROC**
Bagging	0.60	0.49	0.16	0.64	0.65
Voting	0.62	0.50	0.16	0.60	0.65
Stacking	0.62	0.50	0.17	0.61	0.65

For final evaluation phase, Random forest in combination with data balancing using random undersampling has been chosen because of achieved results, the speed and the simplicity (in comparison to other algorithms). Even though ensemble methods achieved almost the same results, Random forest is preferred

because of it's simplicity. In the final evaluation phase, firstly feature selection was performed using *Recursive Feature Elimination with Cross-Validation*. The process of finding optimal number of features is shown in figure 3. As can be seen at the plot, majority of features has been chosen (49 out of 54 features).

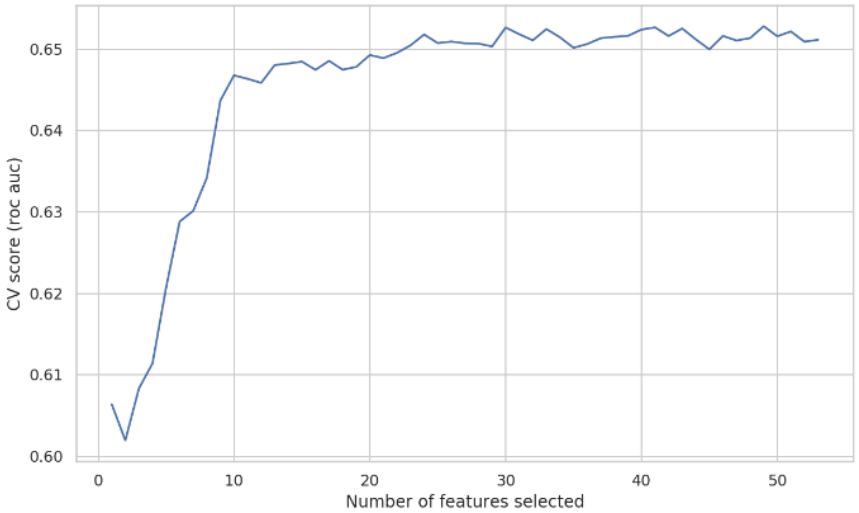


Fig. 3: Finding optimal number of features with RFECV.

After feature selection, hyperparameters of Random forest were optimized with *Random search*, while using only best-selected features from feature selection. Final model with optimized hyperparameters achieved ROC AUC equal to 0.65, that was achieved also without parameters optimization. The learning curve of final model with optimized hyperparameters and using only selected features is shown in figure 4. From the learning curve we can conclude, that adding more data and features may help to achieve even better results. The confusion matrix and ROC curve of final model is shown in figure 5.

Any of the previous efforts to boost up performance of baseline model were not successful. The problem may stem from the nature of the data. Some complementary experiments focused on dividing data to subsets were performed - either using natural subsets existing in the data or using clustering.

As a first approach, data were divided according to age into four age intervals. Then, four individual models (again RF) were trained on samples from corresponding interval. Then, the new samples from test set are predicted with appropriate model according to their age. No significant improvements have been achieved, but at least new information about the nature of the data were observed. Results from individual age intervals showed trend, that performance of model is decreasing with increasing age. Simply, older patients are harder

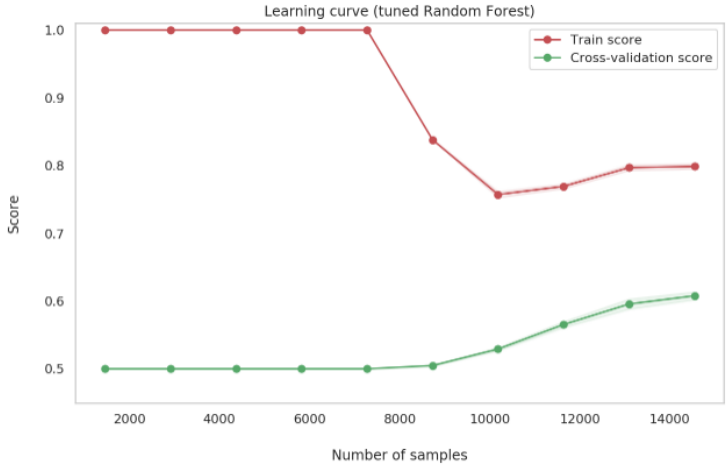


Fig. 4: Learning curve of final model - Random forest with optimized hyper-parameters. Model was trained on balanced data using random undersampling with only features selected by RFECV.

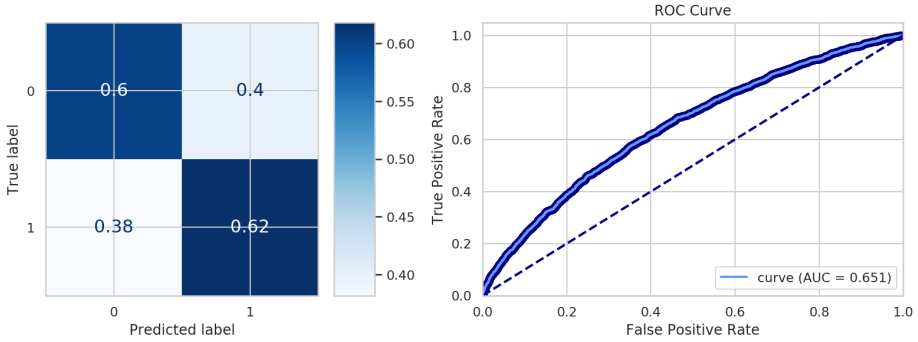


Fig. 5: Confusion matrix and ROC curve of final model.

to be predicted. This observation was also mentioned in several related works. For patients under age of 50 and also at age from 50 to 70, models achieved significantly better results in comparison to model trained on all samples (not divided by age). However, models trained on older patients decreased the overall performance.

In the second complementary experiment, data were clustered using KMeans clustering algorithm. Within each individual cluster, one model was trained. The unseen sample from test set is firstly classified to one of the clusters, then the label is predicted with appropriate model. Again, no improvements were observed.

7 Conclusion

In this work, we were trying to predict diabetic patients early readmission using data-mining approach. To train the machine learning model for this problem, we have chosen popular dataset of diabetes patients clinical encounters across 130 hospitals in United States. Based on the comprehensive data analysis, we have preprocessed the data and create new features. In model selection phase, experiments with Random forest, XGBoost and Multi-layer perceptron were conducted. Each algorithm was trained on three data setups - original imbalanced data, the data balanced with random undersampling and oversampling using SMOTE method. Random forest turned out to be the best performing, so was chosen for final optimization and evaluation. Feature selection using Recursive Feature Elimination with Cross-Validation and hyperparameters optimization using Random search were performed to obtain final model. The final model was trained on undersampled balanced data and achieved ROC AUC score equal to 0.65. Multiple complementary experiments were conducted to boost up the score - ensemble models or dividing data into subsets by either age or clustering. However, no significant improvements were observed. We also analyzed several of related papers where data balancing techniques (SMOTE mostly) boosted the performance of model up to more than 90 percent ROC AUC. 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 instead of real classification problem, that we consider incorrect.

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