

Usage of Gradient Boosting and resampling to predict frauds for medical data

Khasianov Rasul

Sunday 30th September, 2018

Abstract

We aimed to develop a model for detecting cases of prescription fraud and test it on real data from insurance company. We show the proposed baseline works considerably well with a ROC AUC 0.85 and PR AUC of 0.11 for the fraudulent medical prescriptions. Incorporating such model in insurance companies would improve efficiency of fraud detection.

1 Introduction

One of the most important problems of the insurance industry is fraud which causes substantial losses. Fraud detection, being part of the overall fraud control, automates and helps reduce the manual parts of a screening/checking process. This area has become one of the most established industry/government data mining applications.

Data mining tools and techniques can be used to detect fraud in large sets of insurance claim data. Based on a few cases that are known or suspected to be fraudulent, the anomaly detection technique calculates the likelihood or probability of each record to be fraudulent by analyzing the past insurance claims.

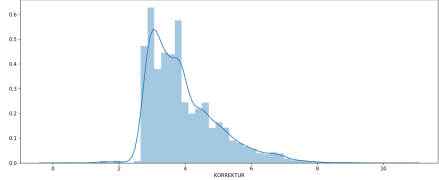
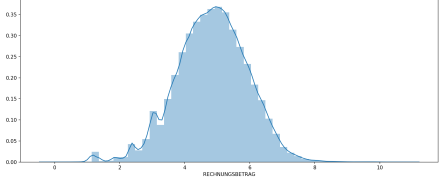
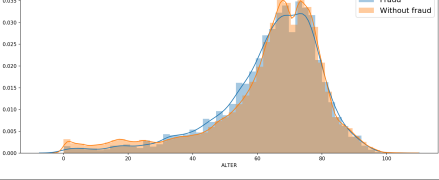
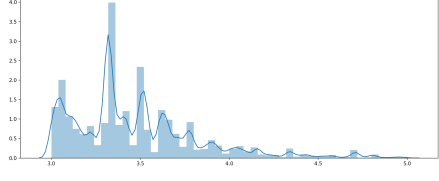
Our problem of fraud detection formulated as a binary classification problem with imbalanced classes: fraud and non-fraud.

Database of a Allianz insurance company was used in this research. There were 1 mln records with 301453 unique receipts. There were 1.75% of fraudulent data and 98.25% of pure data.

Fraud detection analysis was performed by Gradient Boosting: lightGBM [7] and XGBoost [8] algorithms. To balance these 2 classes were used resampling methods such as over-sampling (SMOTE [2], ADASYN [3]), under-sampling (EditedNearestNeighbours [4], [5], InstanceHardnessThreshold, [6]) and their combinations and ensembles.

2 Description of dataset

This database has 4 files with 1 mln records in each file and 13 features. We have trained our models for the first file, that has 301453 unique id.

Name	Meaning	Information
ID	Unique id per receipt	301453 unique id, about 30% of data and 5280 fraud id, about 1.75% of the number of unique values.
KORREKTUR	Target	
RECHNUNGSBETRAG	Invoice amount per document	
ALTER	Age of the customer	
GESCHLECHT	Gender of the customer	0/1-coded: about 48% and 52% respectively
VERSICHERUNG	Type of insurance of the customer	0/1-coded (full or supplementary insurance): about 1% and 99% respectively
FACHRICHTUNG — here problems with data. The same with VERSICHERUNG.	Doctor's specialty	General practitioner, dermatologist, ophthalmologist, etc. (mapped to anonymous values) per document:
NUMMER	Fee number describing the treatment per line	Mapped to anonymous values. 1940 different treatments
NUMMER KAT	Upper category of the fee number per line	Mapped to anonymous values. 17 different categories.
ANZAHL	Number of treatments per line	
FAKTOR	Increase factor of treatment per line	
BETRAG	Cost of treatment per line	
ART	Material cost type per line	Mapped to anonymous values. 11 different types

TYP	Special type of billing per line	0/1/nan-coded: about 90%, 1%, 9% respectively
LEISTUNG	Type of benefits per line	Mapped to anonymous values. Grouping of treatments into collectively agreed types of benefits. 24 different types.

3 Metrics

To evaluate the quality of the algorithm, it is necessary to determine the metrics that will effectively reflect the learning curve of the algorithm on data having an unbalanced nature. Table 3 shows the confusion matrix. Obviously, in the task of fraud detection it is very important to find all actual positive events. Therefore, the lower FN is in relation to TP, the better we will identify cases of fraud. Moreover we can additionally require a small number FP (otherwise this will require unnecessary efforts to verify wheather the event was positive or not)

	Actual positive	Actual negative
Predicted positive	True Positive (TP)	False Positive (FP)
Predicted negative	False Negative (FN)	True Negative (TN)

Table 3: Confusion matrix

Let's discuss some obvious metrics in the conext of our task.

1. Recall = True Positive Rate $\frac{TP}{TP+FN}$. Ideally, this metric should equal 1 and this will mean that we have identified all cases of fraud.
2. Precision = $\frac{TP}{TP+FP}$. We will not be able to demand a high value for this metric. It will tell us what percentage of the predicted values are Actual positive.
3. False Positive Rate = $\frac{FP}{FP+TN}$. It gives us information to events that are not fraudulent. It is minimal when all actual negative events are classified in the correct way.

Let's consider 2 curves:

1. ROC curve. X: FPR, Y: TPR
2. PR curve. X: Recall, Y: Precision

The common ROC curve shows how well both classes are classified. The quality of one class is estimated by the high value of TPR, and the quality of the second class is estimated by the low value of FPR. Therefore, a good prediction is the "balance" between these values.

PR curve characterizes how well one class (fraud class) is well classified. We want to maximize the value of TP in relation to predicted positive and to actual positive. This curve more appropriate for imbalanced dataset. Since the precision-recall plot changes depending on the ratio of positives and negatives, then it is more informative than the ROC plot when applied to imbalanced datasets.

Hence, we can define 2 scores: ROC AUC and PR AUC. AUC is an area under curve.

To choose the best threshold for deviding into 2 classes we will use the maximum of F1-score. F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution (large number of Actual negatives).

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

4 Feature engineering

In the dataset there are 4 features, which are unique for each ID and 8 features that have different lengths of data for each ID. Below are given the descriptions of transformations of the last 8 features.

Feature	Transforming
BETRAG (with ANZAHL)	1. Simple statistics as mean, std, median, min, max 2. Histogram of logarithm of Betrag with 100 bins on ineterval [2, 6] (analogous to the bag of words, just for real value).
FAKTOR, TYP	Simple statistcs as mean, std, median, min, max
NUMMER, NUMMER KAT, LEISTUNG, ART	Bag of words for categorical data

Table 4: Engineering of hierachical features

5 Results and comparison

The thesis [1] is devoted to exactly the same problem and the same data. They prepare data in three ways and fit them with an autoencoder: almost on raw data: only NUMMER, NUMMER KAT, LEISTUNG were encoded with one-hot-encoding (1), on slightly converted data: NUMMER, NUMMER KAT, TYPE, NUMBER, FACTOR, AMOUNT and PERFORMANCE – although by code this is not true, apparently (2), on the data that used the bag of words approach of converting categorical data: NUMMER, NUMMER KAT, LEISTUNG – but here the code is rather strange (3).

In Table 5 we compare quality of models constructed using different methods. Gradient boosting based approaches LightGBM and XGBoost outperfrom approaches based on Autoencoders. As the problem at hand is imbalanced, resampling approaches slightly improve quality of model.

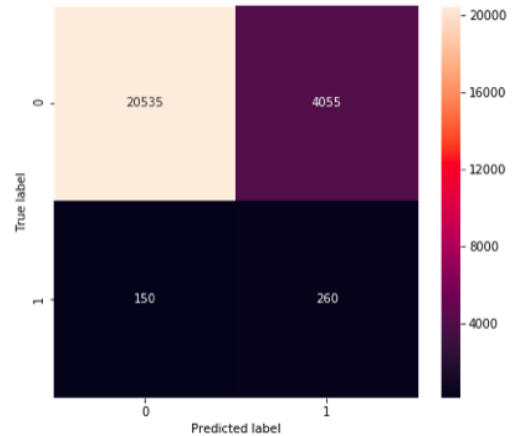


Figure 1: Confusion matrix

Figure 1 shows the distribution of predicted and true labels by confusion matrix. But such matrix depends on threshold value, which defines whether class belongs to fraud or not.

Model	Resampling	ROC AUC	PR AUC
LightGBM	—	0.8345	0.0994
	Over-sampling (SMOTE)	0.844	0.0961
	Over-sampling (ADASYN)	0.8455	0.0985
	Under-sampling (RepeatedEditedNearestNeighbours)	0.8439	0.1019
	Under-sampling (InstanceHardnessThreshold)	0.8515	0.1011
	Combination of over- and under-sampling (SMOTEENN)	0.8485	0.1069
	Ensemble of samplers	0.8446	0.1009
Xgboost	—	0.843	0.1077
	Over-sampling (SMOTE)	0.8037	0.0656
	Over-sampling (ADASYN)	0.7985	0.0612
	Under-sampling (EditedNearestNeighbours)	0.8424	0.1115
	Under-sampling (InstanceHardnessThreshold)	0.8437	0.1071
	Combination of over- and under-sampling (SMOTEENN)	0.8165	0.0832
	Ensemble of samplers	0.8434	0.1017
Autoencoder (1)	—	0.69	—
Autoencoder (2)	—	0.76	—
Autoencoder (3)	—	0.79	—

Table 5: Prediction results of data mining algorithms

6 Conclusion

We considered the fraud detection for medical insurance. As the data is of various length for different cases, we use simple heuristics to generate fixed length feature vectors. To construct classifiers we used Gradient Boosting and resampling as the classification problem at hand is im-balanced. We obtain ROC AUC and PR AUC scores as high as 0.8515 and 0.1115 correspondingly. This ROC AUC score is better than that for Autoencoder based model score ROC AUC 0.79. PR AUC score is 6 times better than PR AUC score for the random classifier.

References

- [1] Philipp Rohde. *Autoencoder for Fraud Detection: An Empirical Application*. Institut für Statistik, 2018.
- [2] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer. *SMOTE: Synthetic minority over-sampling technique*. Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

- [3] H. He, Y. Bai, E. A. Garcia, S. Li. *ADASYN: Adaptive synthetic sampling approach for imbalanced learning*. In Proceedings of the 5th IEEE International Joint Conference on Neural Networks, pp. 1322-1328, 2008.
- [4] D. Wilson. *Asymptotic Properties of Nearest Neighbor Rules Using Edited Data*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 2(3), pp. 408-421, 1972.
- [5] I. Tomek. *An experiment with the edited nearest-neighbor rule*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 6(6), pp. 448-452, 1976.
- [6] M. R. Smith, T. Martinez, C. Giraud-Carrier. *An instance level analysis of data complexity*. Machine learning, vol. 95(2), pp. 225-256, 2014.
- [7] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 3149-3157.
- [8] T. Chen and C. Guestrin. *XGBoost: A scalable tree boosting system*. In Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 785–794, San Francisco, CA, 2016.