Random Forest with High Cardinal, Imbalanced Data

Sina Balkhi
School of Computing Science
Simon Fraser University
sbalkhi@sfu.ca

Saboora M. Roshan
School of Mechatronic Systems Engineering
Simon Fraser university
sabooram@sfu.ca

Max Ou
School of Computing Science
Simon Fraser University
maxo@sfu.ca

Urvi Chauhan
School of Computing Science
Simon Fraser University
uchauhan@sfu.ca

Abstract

Real-world datasets are usually comprised of numerous categorical features which have high-cardinality for one-hot encoding or other common categorical encoding methods to be effective. In the case of high-cardinality, such techniques create large sparse matrices that lead to curse of dimensionality. Furthermore, these datasets also contain class imbalances that adversely affect a classification model's prediction power because the model will be biased towards the majority class and, thus, will not be able to generalize well on new unseen data. In this project, we compare different permutations of categorical encoding methods, namely Target, LeaveOneOut, and Catboost, with two of the most popular over-sampling methods, namely SMOTE and ADASYN, and evaluate each combination by training a random forest classifier. We choose the movie dataset as our case study because it has both high-cardinal features as well as imbalanced classes.

1 Introduction

1.1 Background

An imbalanced dataset refers to the case where the number of instances of one class dominates the number of instances of one or more of the other classes [1]. Machine learning algorithms generalize poorly on unseen data when the dataset used to train the model was significantly imbalanced. The main reason is that the trained model's decision is biased towards the majority class, and so, the minority class may be misclassified.

There are primarily two approaches to handling imbalanced datasets. *Undersampling* is when samples from the majority class, mostly through random choice, are eliminated so that all classes are more or less balanced [2]. However, undersampling may not always be a feasible option if the size of the data is small. Additionally, undersampling is not a precise method for handling such datasets because it may inadvertently discard samples from the majority class that contain important information. In such cases, new samples of the minority class are created using various methods to balance the dataset in what is referred to as *oversampling*. Because our dataset in this project is also small, we will only be focusing on oversampling methods.

Random over-sampling, Synthetic minority over-sampling technique (SMOTE), and ADASYN are common over-sampling methods [5]. However, in many cases, random over-sampling is not practical as it causes overfitting by duplicating some of the original samples [6]. Therefore, a new technique was introduced by Chawla et al. (2002) called Synthetic Minority Over-sampling (SMOTE) [7]. In this method, artificial samples are generated between each sample from the minority class

and its k-nearest neighbours. Another over-sampling method which is introduced by He et al. (2008) is Adaptive Synthetic (ADASYN) which is motivated by the SMOTE technique. ADASYN defines higher ratios for minority class samples that are harder to learn compared to the samples that are easier to learn [8]. Another common method for dealing with imbalanced datasets called weight balancing assigns higher weights for the minority class samples [9]. In this project, we will be using these three methods to evaluate our model's performance in predicting movie success.

However, we do not focus merely on imbalanced datasets. Since there are barely any papers that study the case of imbalanced datasets that also have categorical features with high-cardinality, we also explore how model performance is affected with over-sampling carried out after the categorical features are encoded. Encoding categorical features is tricky because there always exists the risk of information loss after converting a category to a numeric value. For this reason, one hot encoding is a popular and accepted method of encoding where for each category, a new binary feature indicating it is created. However, when the data has high-cardinal features, i.e. large number of categories, e.g. "User ID" , one hot encoding creates infeasibly large amount of new features.

When a feature has high-cardinality, an effective technique of encoding is to group categories by target statistics that estimate expected target value in each statistic. Target encoding[1], a straightforward Bayesian approach, is one such method where a category x_i^m of the ith training example for feature m is estimated as the average value of target y over all the training examples [11]. Both LeaveOneOut and Catboost/ordered encoding are variations of Target and were created to address the target leakage issues inherent in the calculation of Target encoding. We chose these three encodings for our project because all three are comparable to each other and were created for the sole goal of dealing with high-cardinal features for classification tasks.

The comparative study in this report introduces a better understanding of the effects of different sampling and encoding methods in a classification model's performance. There are many sampling methods in literature to deal with imbalanced classes in a dataset. The most well-known method between all of them is SMOTE. To this end, in this project SMOTE and two other methods along with three encoding algorithms are compared with each other for predicting movie success. Moreover, the state of the art interpretation technique LIME is introduced here to evaluate our choice of encoding and sampling techniques.

The rest of the report is organized as follows: Section 2 briefly explains materials and methods including preparing datasets, sampling methods, and encoding techniques. The results and discussion of the comparisons are presented in Section 3, while Section 4 concludes the report.

1.2 Related Work

Bach, M., et al. [4] implemented different over-sampling methods and introduced SMOTE as the best method for osteoporosis patient's prediction. SMOTE was proposed by Colak, M. et al. as the best method for predicting atrial fibrillation in obese patients [9]. Spelmen, V. S. Porkodi, R. [2] compared different sampling methods in their survey and introduced SMOTE as a method that performs better in various applications. A new sampling technique is proposed by Zhur, T. et al. [10] to handle ordinal imbalanced dataset which is called Synthetic Minority Ordinal Regression (SMOR). In [11], SMOTE and k-means SMOTE are compared for 12 imbalanced datasets from UCI Machine Learning Repository. A comprehensive analysis of SMOTE technique is performed in [4] and in this study, SMOTE is suggested for imbalanced datasets. Another comparison was conducted by Fan X. at al. [12] between ADASYN, SMOTE, and Borderline-SMOTE. Chen, C., et al. [7] proposed weighted random forest to penalize misclassifying the minority class and compared the results of weight rebalancing with SMOTE method. Pargent F. [13] compared and benchmarked 10 different encoding methods for high cardinality features with 5 different machine learning algorithms. He found that using the target encoder, which combines simple generalized linear mixed models with cross-validation, shows the most promising results.

Prokhorenkova L.'s team [18] proposed permutation-driven ordered boosting with ordered TS and innovative algorithm for processing categorical features to minimize prediction shifts and target

[1] In this paper, we refer to the encoding method *Target* with the first letter capitalized, as opposed to the *target* variable y.

leakage problems in many existing gradient boosting algorithms. The proposed algorithm is also implemented in the CatBoost gradient boosting library.

2 Data and Methodology

2.1 Data collection and preparation

Our choice of the movie dataset was motivated by the questions we attempted to answer in this project, namely multiple high-cardinality features and imbalanced classes. After cleaning up the dataset by removing missing values and duplicate entries, our final dataset consisted of 2697 movies with 34 features. The main part of the data was collected from The Movie Database API [14]. The features collected from here consisted of cast, director, producer, production companies, genres, release date, tagline, and synopsis. For each movie, we scraped production budget and worldwide gross revenue from The Numbers [15]. We also downloaded movie keywords from Kaggle [16].

We also carried out some feature engineering: We used the "release date" feature to create release month and release day in order for our models to capture any seasonality if it exists. We dropped synopsis and tagline, but used them to create two new "word count" features for each of them. We added "movie keyword" because this represents a categorical feature that will need to be encoded. We did this to aid us in our interpretation of model predictions. For example, assuming movie keywords play little role in determining the success of a movie, if a certain model showed high feature importance for "keyword", this would be indicative of target leakage by our encoder as it means the model is simply memorizing the data.

We define a "success" movie as one where its worldwide gross revenue exceeds its production budget.

$$y_i = \begin{cases} 1 & \text{if production budget} > \text{worldwide gross revenue;} \\ 0 & \text{otherwise} \end{cases}$$

where y_i is the target/response variable for the ith movie.

2.2 Oversampling

2.2.1 SMOTE

In this method, the first step is to compute k nearest neighbors of each minority class sample x_i. Next, a line segment is drawn in the feature space between x_i and its randomly-chosen k nearest neighbors. The minority class is, then, oversampled by introducing synthetic example along these line segments by the equation

$$x_{new} = x_i + (\hat{x}_i - x_i)\delta \tag{1}$$

where $\delta \in [0, 1]$ and is a random number [7].

2.2.2 ADASYN

This technique defines a ratio for generating new samples. The first of this method is calculating the degree of class imbalance ($d \in [0, 1]$), and if $d < d_{th}$ then the algorithm will go to the next step. Here, d_{th} is a threshold for the maximum tolerated degree of class imbalance ratio. Afterwards, the algorithm will go through the following steps:

$$d = \frac{m_s}{m_l} \tag{2}$$

$$G = (m_l - m_s)\beta \tag{3}$$

$$r_i = \overline{K} \tag{4}$$

$$\hat{r}_i = \frac{r_i}{\sum_{i=o}^{m_s} r_i} \tag{5}$$

$$g_i = \hat{r}_i G \tag{6}$$

In these equations, m_s and m_l are the number of minority and majority class samples, respectively, G is the number of minority samples that need to be generated, $\beta \in [0, 1]$ is a parameter to specify the desired balance level, $r_i \in [0, 1]$ is ratio, $_i$ is number of majority samples in the the K neighbors of x_i, $\hat{r}_i$ is a density distribution, and g_i of each minority sample x_i.

In the last step, for each g_i of each minority sample x_i, a loop from 1 to g_i is done in which, one random minority sample x_{zi} of x_i neighbors is selected and then finally a new sample is generated by the equation

$$s_i = x_i + (x_{zi} - x_i)\lambda \tag{7}$$

where $\lambda \in [0, 1]$ is a random number and $x_{zi} - x_i$ is the difference vector in the n-dimensional space [8].

2.3 Categorical encoding

2.3.1 Target

In Target encoding, each distinct categorical sample x_i of a given feature is replaced by the average of the corresponding y values over all training examples. This estimate is usually smoothed by some prior p:

$$\hat{x}_i^k = \frac{\sum_{j=1}^{n} J_{\{x_j^i = x_k^i}.y_j + ap}{\sum_{j=1}^{n} J_{\{x_j^i = x_k^i} + a}, \tag{8}$$

where $a > 0$ is a parameter, x_k^i is a target statistics and y is the response variable. To address target leakage, we smooth our computed means over the training examples with a weight of 200.

2.3.2 LeaveOneOut

LeaveOneOut is a variation of Target where the response y_i of the x_i variable in the ith example being considered is *left out* while calculating the mean of the response y over all training examples. In our experiments, we add "noise" corresponding to standard deviation from a Gaussian distribution while encoding our categorical features with this method to reduce *target leakage*.

2.3.3 CatBoost

Catboost refers to the *ordered* categorical encoding method introduced by Prokhorenkova et al. (2018). Again, this is similar to Target encoding. However, Prokhorenkova et al. consider this to be an effective strategy to reduce target leakage. CatBoost relies on the ordering principle: an artificial "time", i.e., a random permutation σ of the training examples is used to calculate the mean over all y. Although this means we do not need to explicitly add noise, for the sake of more apt comparison with LeaveOneOut, we additionally add noise. This allows us to further reduce any target leakage during model training.

3 Results and discussion

We first split the data to two-thirds train set and one-third test set. Afterwards, we encode each set separately and for each encoding, we applying class weight rebalancing followed by

our two oversampling methods on the train data *only*. For each permutation, we fit a random forest classifier with hyperparameters chosen through cross-validated, randomized search. Due to space constraints in this report, we refer the reader to the online page for this report `https://sinablk.github.io/cmpt726` for all the confusion matrices, features importance and permutation plots, and LIME interpretation results for each iteration of our experiments.

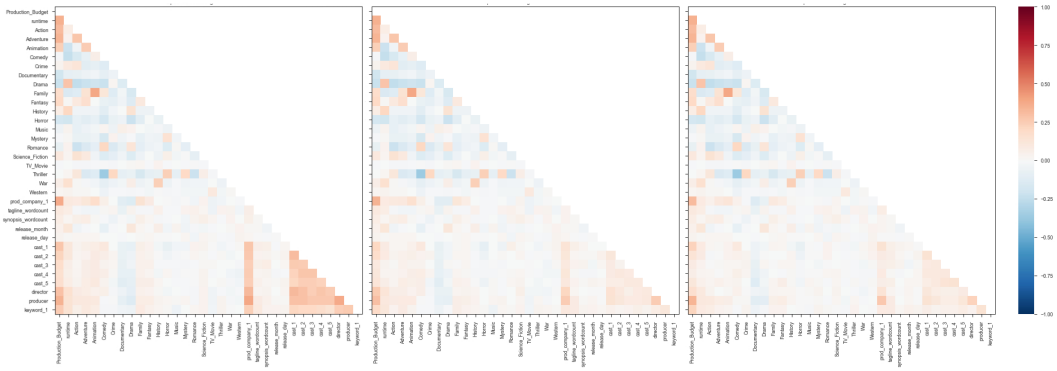

Figure 1: Correlation plots after encoding categorical feature. (*Left*) Target. (*Center*) LeaveOneOut. (*Right*) Catboost. Please visit the link at this page's footnote for high-resolution versions of these plots.

Figure 1 show correlations between the 34 features after the categorical columns were encoded. Our choice of encodings show that our all Target, LeaveOneOut, and CatBoost correctly captured the relations between features. For example, cast members, producers, directors, and production companies are more correlated with each other than other features. Because genres do not represent high cardinal features, in contrast to other categorical features, we used one-hot encoding for this feature. As expected, one hot encoding also seems to capture feature relationships correctly. For example, `Family` has relatively high correlation with `Animation` while `Horror` is highly uncorrelated with `Comedy`. Interestingly, categorical features that are encoded using Target exhibit higher collinearity than the other two.

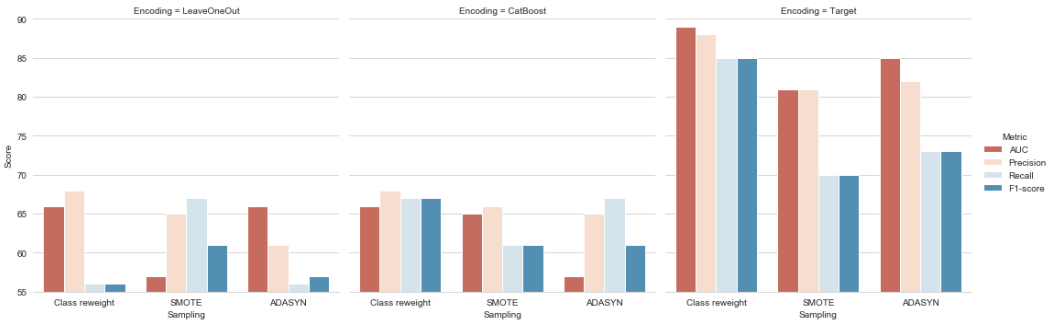

Figure 2: Comparison between different sampling and encoding methods with decision tree.

To assess the performance of our models, we use a number of classification metrics to get a better sense of how our choice of oversampling and encoding affect the performance of a tree-based model. In this paper, we will not discuss in detail about what each classification metric represents, but simply state that they are appropriate metric in classification tasks and for each metric, the higher it is, the more accurate our model's predictions. For a details of each metric, we refer the reader to literature on the web. All results here represent the prediction on unseen test data.

Figure 2 shows our experiments with a single decision tree. Although the subject of our project was binary classification with random forest, we still attempted to see if a single decision tree could perform better than an ensemble of trees. However, we were also motivated to use a decision tree to assess our oversampling and encoding techniques because of the interpretability that a single

decision tree offers. In Figure 2, we can see that Catboost encoder outperforms our other choices of encoding and oversampling, while Target, even after smoothing, clearly suffers from target leakage as evidenced from the high test scores.

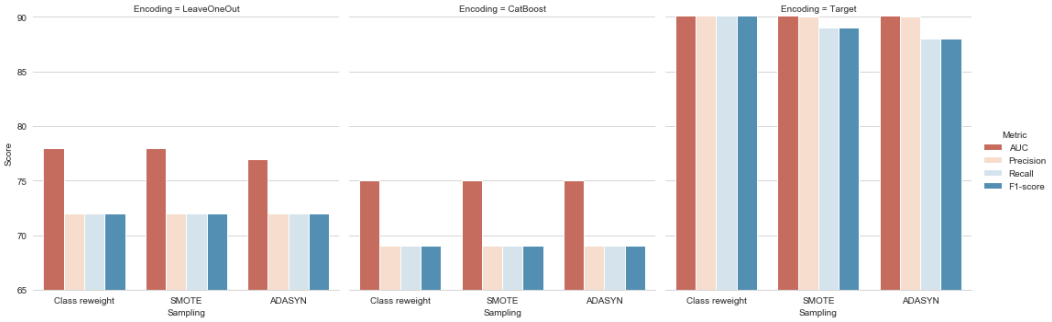

Figure 3: Comparison between different sampling and encoding method with random forest.

Figure 3 shows our model performance results on test data using random forest model. Both our models (decision tree and random forest) show that Catboost encoding method with simple class reweight or SMOTE oversampling outperform other encodings and sampling methods. It must be noted that by *best*, we mean one where target leakage is at a minimum with logical accuracy scores in a data that is both small and "noisy". We do not expect high test scores such as those shown by Target encoding.

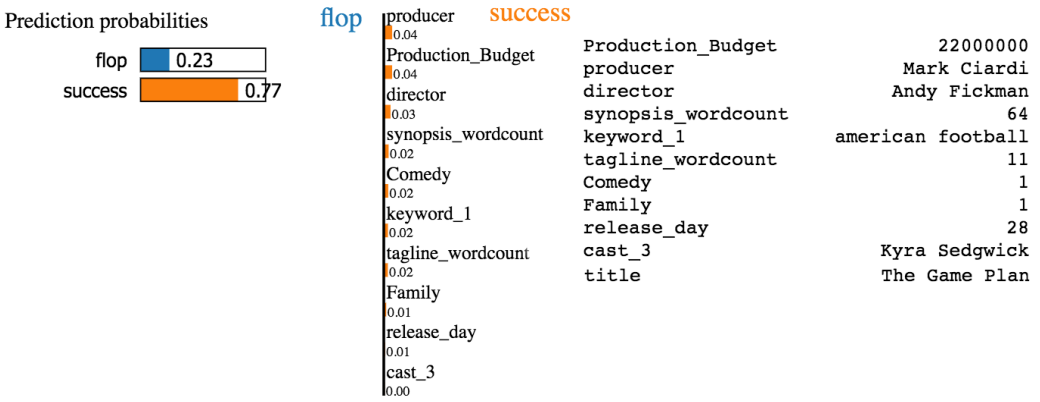

Figure 4: LIME explanation for movie "The Game Plan" with SMOTE over-sampling, CatBoost encoding, and random forest.

Local Interpretable Model-agnostic Explanations (LIME) is a technique that can be applied to any machine learning model. The technique attempts to understand the model by mutating the data input and learning how the predictions change. For more explanation of LIME, we refer the reader to the original paper [17]. We applied LIME on our trained models to further explain our predictions.

Figure 4 shows our LIME interpretation results from a test sample that was classified by a random forest trained with SMOTE oversampling and CatBoost encoding. From the overall outputs, we can see that features such as `producer`, `Production_Budget`, and `director` contributed to our sample being classified as a "success". If target leakage were present, we would see all our encoded categorical variables on the top of that list. Because we can see that our model was able to use non-encoded variables such as `Production_Budget`, `synopsis_wordcount` (which is "noise"), and `Family` is indicative that target leakage is minimized. This was consistent with most of the other samples we interpreted using LIME.

4 Conclusion and Future Scope

In this project, we have shown that categorical encoding of high-cardinal features when the classes in a dataset are highly imbalanced is feasible in classification tasks. Furthermore, we have shown that in such cases, Catboost and LeaveOneOut with added noise minimizes target leakage inherent in such Bayesian encoders that use the response variable y to encode a given categorical variable. We have also shown that the time-tested SMOTE oversampling method is best method when dealing with imbalanced datasets. However, in some cases, a simple weight rebalancing suffices.

In future studies, we will evaluate our results on different datasets, and also we will work on multi-classification.

Contributions

Saboora M. Roshan carried out entire literature review and contributed to major parts of this report. She also designed the poster. Urvi Chauhan researched categorical encoding methods and authored the **Encodings** part of this report as well as its corresponding paragraph under **Related work**. Max Ou interpreted the results of our trained models using LIME and authored significant portions of this report. Sina Balkhi carried out model training with random forest and decision trees and all related work with model training and evaluation, and authored the corresponding portions of this report.

References

[1] Rout, N., Mishra, D. Mallick, M. K. (2018) Handling Imbalanced Data: A Survey. In: Reddy M., Viswanath K., K.M. S. (eds) International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications. Advances in Intelligent Systems and Computing, (628), Springer, Singapore, pp. 431-443.

[2] Spelmen, V. S. Porkodi, R. (2018) A Review on Handling Imbalanced Data, 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, India, pp. 1-11.

[3] He, H. Garcia, E. A. (2009) Learning from Imbalanced Data, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 21(9), pp. 1263-1284.

[4] Bach, M., Werner, A., Zywiec, J. Pluskiewicz, W. (2017) The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis, Journal of Information Sciences, (384), pp. 174-190.

[5] Chawla, N. V., Bowyer, K. W., Hall, L. O. Kegelmeyer, W. P. (2002) SMOTE: Synthetic Minority Over-Sampling Technique, Journal of Artificial Intelligence Research (16), pp. 321–357.

[6] He, H., Bai, Y., Garcia, E. A. Li, S. (2008) ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, 2008 IEEE International Joint Conference on Neural Networks, Hong Kong, China, pp. 1322-1328.

[7] Chen, C., Liaw, A. Breiman, L. (2004) Using Random Forest to Learn Imbalanced Data. Statistics Department, University of California, Berkeley, California. Technical Report 666.

[8] Micci-Barreca D. (2001) A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems, ACM SIGKDD Explorations Newsletter, 3(1): 27-32.

[9] Colak, M. et al. (2017) Handling imbalanced class problem for the prediction of atrial fibrillation in obese patient, Biomedical Research, India, (28) pp. 3293-3299.

[10] Zhur, T. et al. (2019) Minority oversampling for imbalanced ordinal regression, Journal of Knowledge-Based Systems (166) pp. 140–155.

[11] Douzas, G., Bacao, F. Last, F. (2018) Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE, Journal of Information Sciences, (465), pp. 1–20.

[12] Fan X., Tang K., Weise T. (2011) Margin-Based Over-Sampling Method for Learning from Imbalanced Datasets. In: Huang J.Z., Cao L., Srivastava J. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2011. Lecture Notes in Computer Science, vol 6635. Springer, Berlin, Heidelberg.

[13] Pargent F., Bischl B., Thomas J. (2019) A Benchmark Experiment on How to Encode Categorical Features in Predictive Modeling. Retrieved from https://osf.io/356ed/download, December 1st, 2019.

[14] Available on: https://developers.themoviedb.org/3, accessed: Nov. 2019.

[15] Available on: https://www.the-numbers.com/movie/budgets/all, accessed: Nov. 2019.

[16] Available on: https://www.kaggle.com/rounakbanik/the-movies-datasetkeywords.csv, accessed: Nov. 2019.

[17] Ribeiro, M. T., Singh, S. Guestrin, S. (2016) "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 1135–1144.

[18] Prokhorenkova L., Gusev G., Vorobev A., Dorogush A. V., Gulin A. (2017) CatBoost: unbiased boosting with categorical features. Retrieved from https://arxiv.org/abs/1706.09516, December 4th, 2019.