Comparison of Naive Bayes and Tree-based Classifiers for Imputing Class Imbalanced Healthcare Data

Nathan Zencey and David Robison

Abstract

As the volume of data available for analysis continues to increase, the need for more powerful data imputation methods also increases. At scale, poor data imputation practice may introduce significant bias for purposes of learning. We compare the performance of several classifiers to naive methods for imputing missing data in the Healthcare Cost and Utilization Project (HCUP) State Inpatient Databases (SID). Our findings describe adjustments for class imbalanced classification problems and the strength of random forest decision tree learners, which showed best cross validated performance across 4 of 5 response variables.

Introduction

Current data mining efforts in healthcare present statistical challenges that require treatment of data to maintain the validity of findings. For example, large patient datasets are often affected by spurious or missing data across a population[1]. Given the prevalence of missing data, significant attention has been given to data imputation methods with the literature defining three types of missing data: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR)[2].

The State Inpatient Databases (SID) is a subset of HCUP, and contains inpatient discharge data from community hospitals in 28 states. SID suffers from moderate missing data in several patient level variables such as race or admission source. Past work on SID has described the importance of assessing patterns in the data to first learn the missing data. As such we review the distribution of our five categorical response variables: race, income level, hospital admission type, admission source, and total charge of the visit[3].

As shown below in Table 1, the three response variables of RACE, ATYPE, and ASOURCE show significant class imbalance defined as one class equaling greater than 50 percent of the distribution. Similar to the challenge of missing values in data as discussed above, classification for multi-class imbalanced responses requires attentive treatment[4].

---

[1] Hersh, William R. et al. “Caveats for the Use of Operational Electronic Health Record Data in Comparative Effectiveness Research.” Medical care 51.8 0 3 (2013): S30–S37. PMC. Web. 30 Oct. 2017.
[2] Little, Roderick J. A., and Donald B. Rubin. Statistical Analysis with Missing Data. Wiley, 2009.
[3] Ma, Yan, et al. “The HCUP SID Imputation Project: Improving Statistical Inferences for Health Disparities Research by Imputing Missing Race Data.” Health Services Research, Apr. 2017. ePub Ahead of Print.
[4] Hulse, Jason Van, et al. “Experimental Perspectives on Learning from Imbalanced Data.” Proceedings of the 24th International Conference on Machine Learning - ICML '07, 2007, doi:10.1145/1273496.1273614.

Figure 1: Distribution of Race, Type, Total Charge, Admission, and Income Quartile

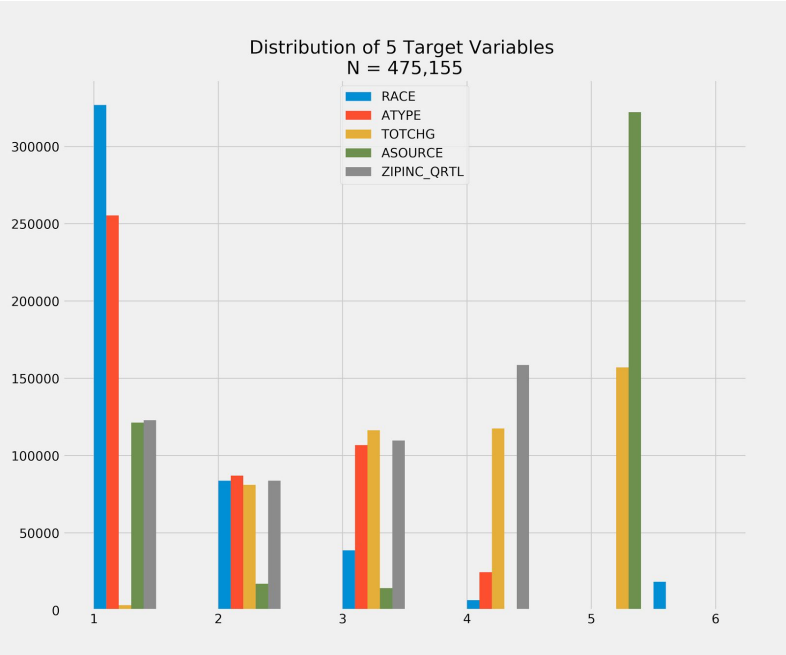


Table 1: Percent Distribution of Race, Type, Total Charge, Admission, and Income Quartile

	1	2	3	4	5	6
RACE	68.8%	17.6%	8.16%	1.38%	0.15%	3.87%
ATYPE	53.7%	18.3%	22.5%	5.2%	0.15%	0.15%
TOTCHG	0.67%	17.1%	24.5%	24.7%	33.1%	NA
ASOURCE	25.5%	3.6%	3.0%	0.02%	67.8%	NA
ZIPINC	25.9%	17.6%	23.1%	33.4%	NA	NA

Approaches for classification under class imbalanced circumstances have been described in many applications such as fraud detection[5] and diagnosis of rare disease[6]. The literature consistently makes two recommendations for imbalanced data: cost sensitive learning, where a high cost is assigned to misclassified minority classes, or resampling the data to down-sample the majority class, oversample the minority class, or both[7].

To account for the class imbalanced response variables, we assess our classifiers against multiple classification metrics:

[5] Fawcett, T. E. & Provost, F. (1997). Adaptive fraud detection. Data Mining and Knowledge Discovery, 1, 291–316.
[6] Murphy, P. M. & Aha, D. W. (1994). UCI repository of machine learning databases. University of California-Irvine, Department of Information and Computer Science. http://www1.ics.uci.edu/mlearn/ MLRepository.html.
[7] Chen, Chao and Andy Liaw. "Using Random Forest to Learn Imbalanced Data." (2004).

- **Accuracy score**, defined as the ratio of correctly classified observations to total observations
- **Precision (P)**, defined as $P = T_p / T_p+F_p$ where T_p is the number of true positives and F_p the number of false positives, and
- **Recall (R)**, defined as $R = T_p / T_p+F_n$ where T_p is the number of true positives and F_n the number of false negatives.
- **F1 score,** the mean of precision and recall for a given class defined as $F1 = \frac{2(P * R)}{(P + R)}$

High scores for precision and recall indicate a classifier that returns accurate results (high precision) and the majority of all positive results (high recall). Finally, we also explore the effect of resampling[8] for the imbalanced response variable of Race using synthetic minority oversampling technique (SMOTE) and random under sampling.

For this analysis, we use explore three supervised learning algorithms: Bernoulli Naive Bayes, Random Forest, and Gradient Boosted Tree classifiers. Below, we review the theoretical basis of learning algorithms. Afterwards, we describe our data preprocessing steps and review cross-validated classifier performance scores.

Models Background

Naive Bayes

The Naive Bayes classifier is a supervised learning algorithm that imposes "naive" assumptions on Bayes' theorem, and considers all predictor variables $x \in X$ as independent[9]. Using the naïve assumption, Bayes theorem is re-expressed as:

$$P(y \mid x_1, x_3, ..., x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x1, x2, ..., xn)}$$

Often and despite this restrictive assumption, a Naive Bayes model will perform extremely well on classification tasks without the need for parameter tuning or significant computational power. As such, we use a Bernoulli Naive Bayes classifier with binary features as a baseline classifier. The Bernoulli Naive Bayes updates the decision rule[10] for Bayes Theorem to the following:

$$P(x_i \mid y) = P(i \mid y) x_i + (1 - P(i \mid y)(1 - x_i)$$

Decision Tree Methods

Decision tree learners are a form of non-parametric supervised learners methods. Decision trees work using the concept of creating an ensemble of models by recursively partitioning or making "splits" in the feature space so that like observations are grouped together. Decision trees provides intuitive and highly

[8] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Int. Res. 16, 1 (June 2002), 321-357.
[9] Artificial intelligence a modern approach/ Stuart Russell ; Peter Norvig Stuart Russell, Peter Norvig - Prentice Hall - 2003
[10] Zhang, Harry. (2004). The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004. 2.

interpretable models. The quality of a split is often measured using one of two metrics. In our case, we use *Gini Impurity* for evaluating the information gain with each split where Gini Impurity is:

$$\mathbf{Gini(E) = 1 - \sum_{j=1}^{c} p^2_j}$$

Where j is the class of the target variable and p represents probability. Figure 2, shown below, provides an excellent visualization of a decision tree on the "Iris" dataset[11].

Decision Trees do have some import limitations, showing high variability and and thus prone to overfitting. In our analysis, we use two extension of the Decision Tree algorithm: Random Forest and Gradient Boosted classifiers.

Random Forest

A random forest extends the decision tree classifier by implementing steps to reduces the variance of a single decision tree by multiple trees to build an ensemble. Furthermore, for each tree, a subset of predictor variables are selected at random and only these variables are considered for each split in each tree. By introducing this randomization, the errors of each decision tree are less correlated.

Random forests are far more accurate on test data than a single decision tree. However, as one cannot easily view the attributes used to make splits, or the splits across hundreds of trees, they are considered a "black box" model.

As random forests are not highly interpretable, systematic cross validation of the parameters used to create a random forest (number of trees grown, depth of each tree, number of attributes used to make splits, minimum amount of observations that can be split, minimum size of a leaf) is essential.

Gradient Boosted Trees

Boosting is a form of forward stagewise additive modeling, using sequentially trained weak learners to produce a strong classifier as an ensemble of weak learners.[12] A weak classifier is defined as one where the error rate is only slightly better than random guessing. The predictions of all the weak classifiers are combined through a weighted majority vote to produce a final function of the form:

$$F(x) = sign(\sum_{m=1}^{M} \alpha_m h_m(x))$$

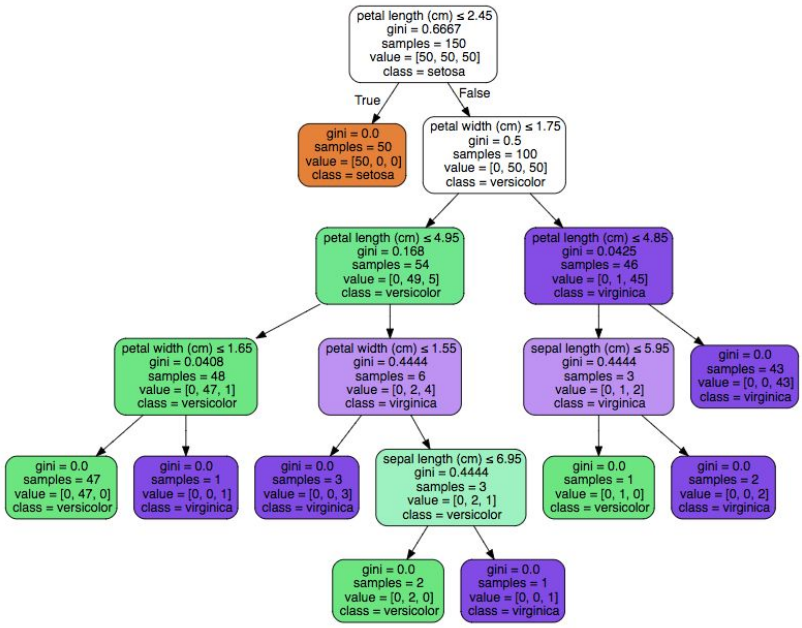
Where $h_m(x)$ are the set of base weak classifiers or learners m = 1,2, …., M and the parameter α_1 α_2, α_m are the weighting for the contribution of each weak learner to the next weak learner in the stagewise additive model:

$$F_m(x) = F_{m-1}(x) + \alpha_m h_m(x)$$

[11] Scikit learn. Decision Trees. Available at: http://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html

[12] Friedman, Jerome H. Greedy function approximation: A gradient boosting machine. Ann. Statist. 29 (2001), no. 5, 1189--1232. doi:10.1214/aos/1013203451.

Figure 2: Example of Decision Tree Classifier Splits Based on Gini Impurity



For each stage, the new weak learner decision tree, $h_m(x)$, is chosen so as to minimize the loss function L given the current ensemble of weak learners , expressed as:

$$F_m(x) = F_{m-1}(x) + arg\ min_h\ L(y_i,\ F_{m-1}(x) + h(x))$$

The optimization of this loss function L requires that it be differentiable so that a negative gradient can be calculated. At each step, a greedy function framework is iteratively used to select the most negative gradient or steepest descent. For a large enough set of base weak learners, the Gradient Boosted Tree will converge to the global minimum, thus optimizing the function and providing less bias estimators.

Methodology

Data Preparation

Prior to model training for the five response variables: RACE, ZIPINC, ASOURCE, ATYPE, and TOTCH we took steps to clean and prepare the source data. Of special importance is the inability of our selected learning algorithms to handle null data. In addition, where needed, we took steps to binarize categorical predictors. For each learning algorithm, prior to splitting our data into testing and training sets, we took the following steps:

1. Remove attributes with more than 90% missing values
2. Remove any rows with missing values
3. Convert discrete values to binary form
4. Partition the dataset into 67% training and 33% testing sets

In order to classify the continuous total charge (TOTCH) response variable, we created five class labels along the range of values defined below:

Class	TOTCHG
1	0 - 1,000
2	1,001 - 5,000
3	5,001 - 10,000:
4	10,001 - 20,000
5	20,000 - 1,500,000

Implementation and Parameter Tuning

We implemented all estimators using Python and its associated scientific computing libraries. To conduct a nested cross-validation of our estimators, we use the Grid Search and Pipeline modules of *sci-kit learn*. For additional hyperparameter tuning of the Gradient Boosted classifier for the Race response variable, we produce a set of validation curves before training a final classifier.

Due to computational limitation, we randomly subsampled the full data set in order to train our estimators. We acknowledge this introduction of sampling bias into the optimized parameter search. Still, we maintain that cross validation performed on a smaller data subset is useful for approximating the proper "neighborhood" of ideal parameters.

To explore resampling, we used several methods in the the Python module *imblearn*. For example, to oversample the minority classes of the Race response variable, we used the Adaptive Synthetic Sampling Approach for Imbalanced Learning (ADASYN) method. ADASYN generates minority class samples using a k-Nearest Neighbors approach, focusing on synthetic samples that are near original samples which were wrongly classified. We also used ADASYN's random undersampling methods on majority classes; random undersampling simple takes a random (and thus representative) slice of the majority class to better balance it with the minority classes.

Results

Results with Imbalanced Data

Table 2 shown below reports the Accuracy Score, Precision, and Recall performance of our models without resampling. As discussed above, comparison with more rudimentary data imputation is a key objective of our analysis. Therefore, Table 2 also provides estimates of a "null" model approach that would impute using the most frequent available label in each class.

Our Random Forest classifiers consistently show best performance when compared with Bernoulli Naive Bayes and Gradient Boosted Trees. The relative poor performance of Bernoulli Naive Bayes likely indicates that the "naive" assumptions do not hold or that the reduced feature space of only binary variable are not strong predictors of any response variable.

Results for Race Variable with Resampled Data

To explore the effect of resampling we chose the class imbalanced Race response variable for experimentation. Table 3 shows our findings where we compare the best model performance (by F-1 scores) obtained for each class in our non-resampled Random Forest Model and class resampled Random Forest. Our goal here was to use resampling in order to improve our in-class F-1 scores, indicating that minority classes were being accurately classified, without minimal reductions in overall accuracy.

Our findings for resampling of the Race variable show mixed results, but point toward majority undersampling as a potential method best way to improve F1 scores for minority classes while maintaining similar performance for majority class classification as judged by accuracy score.

Table 2: Comparison of Model Performances on Classification Metrics

	Model	Accuracy	Precision	Recall	Average F1 Micro
RACE: Null: 68.77%	Naive Bayes	69.05%	0.99	0.69	0.81
	Random Forest	72.83%	0.74	0.73	0.73
	Gradient Boost	**76.84%**	0.74	0.77	0.74
ZIPINC_QRTL Null: 33.33%	Naive Bayes	35.22%	0.65	0.35	0.45
	Random Forest	**62.30%**	0.61	0.62	0.61
	Gradient Boost	60.64%	0.59	0.61	0.60
ASOURCE Null: 67.82%	Naive Bayes	65.75%	0.85	0.66	0.73
	Random Forest	**94.00%**	0.94	0.94	0.94
	Gradient Boost	92.69%	0.92	0.93	0.92
ATYPE Null: 53.74%	Naive Bayes	60.84%	0.67	0.61	0.63
	Random Forest	**89.54%**	0.89	0.90	0.89
	Gradient Boost	87.12%	0.87	0.87	0.87
TOTCHG Null: 33.10%	Naive Bayes	41.58%	0.48	0.42	0.44
	Random Forest	**79.14%**	0.79	0.79	0.79
	Gradient Boost	72.58%	0.73	0.73	0.72

Table 3: Comparison of Random Forest F1-Scores for Race with Class Imbalance and Resampling

Race	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Accuracy
Imbalanced F1-Score (Sample Size)	0.88 (218876)	0.49 (56256)	0.43 (25949)	0.02 (4398)	0.02 (491)	0.36 (12383)	76.50%
Majority Undersample: F1-Score (Sample Size)	0.88 (50000)	0.51 (20000)	0.49 (15000)	0.08 (4933)	0.01 (550)	0.38 (13824)	75.90%
Majority Undersample: F1-Score (Sample Size)	0.85 (10000)	0.53 (6000)	0.47 (3500)	0.12 (2000)	0.07 (550)	0.38 (3000)	72.83%
Minority Oversample: F1-Score (Sample Size)	0.88 (34302)	0.52 (34302)	0.44 (34302)	0.02 (34302)	0.01 (34302)	0.36 (34302)	76.36%

Conclusion

We find in most cases that a random forest is the most effective model. Severe class imbalances present a challenge for classification models. As demonstrated in our results with the resampled Race variable, this can be partially addressed by down sampling or oversampling, however, considerable experimentation is needed to arrive at the optimally resampled data. While optimized ensemble models achieved respectable accuracy and average F-1 scores, their precision and recall performance for extremely rare class labels was poor. This is expected, as ensemble models are designed not to overfit based on rare fluctuations that statistically amount to noise. If a researcher's requires greater recognition of such rare class labels, a mix of ensemble classifiers and anomaly detection models may provide a better learning framework.