Medicare Fraud Detection using CatBoost

John Hancock and Taghi M. Khoshgoftaar
College of Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida 33431
jhancoc4@fau.edu, khoshgof@fau.edu

***Abstract*—In this study we investigate the performance of CatBoost in the task of identifying Medicare fraud. The Medicare claims data we use as input for CatBoost contain a number of categorical features. Some of these features, such as the procedure code and provider zip code, have thousands of possible values. One contribution we make in this study is to show how we use CatBoost to eliminate some data pre-processing steps that authors of related works take. A second contribution we make is to show improvements in CatBoost's performance in terms of Area Under the Receiver Operating Characteristic Curve (AUC), when we include another one of the categorical features (provider state) as input to CatBoost. We show that CatBoost attains better performance than XGBoost in the task of Medicare fraud detection with respect to the AUC metric. At a 99% confidence level (with p-value 0) our experiments show that XGBoost obtains a mean AUC value of 0.7615 while CatBoost obtains a mean AUC value of 0.7851, validating the significance of CatBoost's performance improvement over XGBoost. Moreover, when we include an additional categorical feature (healthcare provider state) in our data analysis, CatBoost yields a mean AUC value of 0.8902, which is also statistically signficant at a 99% confidence interval level (with p-value 0). Our empirical evidence clearly indicates CatBoost is a better alternative to XGBoost for Medicare fraud detection, especially when dealing with categorical features.**

***Keywords*—Gradient Boosted Decision Trees, CatBoost, XGBoost, Class Imbalance, Medicare Fraud**

1. Introduction

This is a study on using the XGBoost and CatBoost machine learning algorithms for Medicare insurance fraud detection. Medicare is the United States Federal Government's health insurance program. The population of individuals eligible for Medicare includes people age 65 and over, people with certain disabilities, and others. Please see [1] for more details on Medicare eligibility requirements. Hence, a large segment of The United States' population qualifies for Medicare. In 2019, Medicare had 61.2 million beneficiaries [2]. Since so many are eligible, the agency responsible for Medicare, the Centers for Medicare and Medicaid Services (CMS), processes a huge volume of requests for payments. These requests are also known as "claims" or "insurance claims," some of which are fraudulent. For example, a fraudulent claim might be for procedures a provider did not actually render to a patient. In 2017, the United States Government Accountability Office (GAO) reported, "Although there are no reliable estimates of fraud in Medicare, in fiscal year 2017 improper payments for Medicare were estimated at about $52 billion [3]." Hence, there is an opportunity in detecting fraudulent insurance claims sent to Medicare. From our point of view as taxpayers, we would like to know what part of the $52 billion in improper payments is due to fraud on the part of healthcare providers. If it is a sizable amount, we as taxpayers would like to see that money better spent.

What sets this study apart from previous work is that this is the first study to employ CatBoost [4] to tackle the challenge of Medicare fraud detection with machine learning. We are motivated by the claims of Prokhorenkova *et al.* in the seminal work "CatBoost: unbiased boosting with categorical features" [4]. In their work, the authors make the claim that tree-based algorithms are better suited for machine learning tasks on heterogeneous data. The Medicare claims data we use in this study are heterogeneous since they consist of a mix of categorical and numerical features. This work builds on previous forays to take on the task of detecting fraud in Medicare insurance claims data [5], [6], [7], [8]. These works take advantage of two data sources. The first source is claims data CMS publishes annually in, electronic form, in the publicly available Medicare Provider Utilization and Payment Data: Physician and Other Supplier Files [9]. The second source is the United States Office of the Inspector General's (OIG's) List of Excluded Individuals and Entities (LEIE) [10]. Our contribution is that we show CatBoost is a competitive candidate in the realm of classifiers of highly imbalanced data. Furthermore, CatBoost handles categorical features automatically; therefore, simplifying the work of preparing data for the classification task. As a result, we also indicate avenues for future research in the employment of CatBoost as a classifier for imbalanced, heterogeneous, categorical data. Although this is the first study (to our knowledge) on using CatBoost for Medicare fraud detection, there exists much previous work on the subject of using machine learning for Medicare fraud detection. Therefore, in the next two sections we provide some

978-1-7281-1054-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IRI49571.2020.00022

background on related work, including some background on XGBoost and CatBoost.

2. Related Work

There exists a body of research on applying machine learning to the problem of detecting fraud in Medicare insurance claims. Here, we recount key publications in the line of work that leads to this study. These previous publications lead us to the question of whether CatBoost, with its support for working with categorical variables of high cardinality, is a suitable algorithm for Medicare fraud detection.

In 2016, Bauder and Khoshgoftaar published a study [11] using a Bayesian probability model to detect anomalies in Medicare data using probabilistic programming. That study proved the effectiveness of machine learning techniques for detecting fraud in the same Medicare claims data we use in this study. The authors show that their Bayesian model flags data related to a provider under investigation for fraudulent activity [12]. This work set the stage for applying machine learning to electronic records to detect suspicious Medicare billing practices.

A related study along the lines of detecting fraud using data from CMS is a paper by Bauder and Khoshgoftaar, [5]. In this study, published July 2016, Bauder and Khoshgoftaar develop a novel technique for detecting fraudulent activity. They fit several multivariate regression models to Medicare claims data from CMS. The authors then compute a vector of expected payments to the provider, using the regression models. They then calculate the errors, or residuals, the models yield, using the actual and expected vectors of payments. When the computed residuals for a provider are high, they treat this as a flag for fraudulent activity. In [5], the authors mention that their technique finds one provider listed in the LEIE. This finding shows the potential for automation to match data on providers from CMS with data on providers in the LEIE.

The technique of automatically matching records in CMS data by NPI to records in the LEIE is described in Branting et al. [8]. In their work, published in August 2016, the authors derive graphs from publicly available data to predict the presence of a provider in the LEIE. The algorithms produce graphs that give a quantitative notion of either provider geospatial similarity or provider behavioral similarity. The authors then use this similarity to predict whether a provider is in the LEIE. Of particular interest to us is the authors' claim that an ablation analysis of their models' features shows that the features most important for accuracy are those related to geospatial co-location. We find a similar lift in accuracy when we include the National Plan and Provider Enumeration System (NPPES) state as a feature in the CatBoost input data.

Another earlier work that leads to our study is [7], by Bauder and Khoshgoftaar. In their article Bauder and Khoshgoftaar report that they use oversampling, and undersampling techniques, and that their undersampling technique creates a more representative dataset than in [8]. One may also notice that in the works we review thus far, there is a mix of supervised and unsupervised learners. In [7], Bauder and Khoshgoftaar also compare the performance of supervised, unsupervised, and hybrid learners in combination with sampling techniques. The authors find that supervised learners and undersampling to an 80-20 imbalance ratio provides the best performance in terms of area under the Receiver Operating Characteristic curve (AUC) [13]. Due to space limitations we do not give details on the AUC metric here. Please see [13] and [14] for more information on AUC.

Since Bauder and Khoshgoftaar find supervised learners to be best suited to the task of identifying fraud in Medicare data in [7], a logical next step is to focus on supervised learners and undersampling techniques. In [15], Bauder and Khoshgoftaar study Random Forest's ability to detect anomalous behavior in Medicare data with various class imbalance ratios. They control the imbalance ratios with random undersampling. We see that in [15], Bauder and Khoshgoftaar report the best performance of Random Forest with a mean AUC value of 0.87302 with a class distribution of 90% of instances labeled non-fraudulent to 10% of instances labeled as fraudulent.

Building on the success of [15], in [16], and [17], Johnson and Khoshgoftaar investigate the performance of deep learning algorithms on the Medicare fraud detection task. In [17], Johnson and Khoshgoftaar expand the scope of sampling techniques, in conjunction with varying the classification threshold of learners. They show that a hybrid technique of random undersampling and random oversampling have an impact on deep learning algorithms' performance in terms of AUC.

In the studies we cover thus far, we notice that researchers have yet to investigate the performance of CatBoost on the task of identifying fraud in Medicare data. In the Medicare data we use for this study, there are several categorical features. One such categorical feature is the Healthcare Common Procedure Coding System (HCPCS) code. This feature has 7,028 possible values in our dataset. The related works [7], [5], do not employ this feature directly. However, CatBoost handles representation of categorical variables internally. This makes using high cardinality categorical features like the HCPCS code relatively easier to use. Hence, we take advantage of the opportunity to use CatBoost to make a contribution to the body of knowledge that these related works also belong to. Our contribution is to compare two different types of gradient boosted decision tree algorithms on the task of detecting fraud in Medicare data. In the next section, we discuss some details on these algorithms.

3. Gradient Boosted Decision Trees

Earlier we noted that Prokhorenkova et al. in [4] claim that ensembles of gradient boosted decision trees (GBDT algorithms) are well-suited to operate on heterogeneous data. Heterogeneous datasets contain features with different data types. Tables in relational databases are often heterogeneous. Medicare claims data from CMS is a specific example of heterogeneous data. Therefore, we are motivated in this

study to investigate the performance of XGBoost [18] and CatBoost [4] as classifiers of Medicare claims data, since both are GBDT algorithms.

GBDT algorithms are a family of variations on an ensemble technique that iteratively build a set of decision trees, to be used collectively, to do classification or regression. We attribute the discovery of GBDT's to Friedman [19]. GBDT algorithms add decision trees to the ensemble by selecting the tree that approximately minimizes the value of the loss function $\mathcal{L}(\hat{y}, y)$ defined on the set of output values of the ensemble, $\hat{y}$, and the labels, y, on some dataset we use for input to the GBDT algorithm. After training, the output probability of the ensemble is the sum of the probabilities for classes indicated in the leaf nodes of the decision trees.

After Friedman's publication on GBDT's, many researchers have contributed related improvements. An example is XGBoost [18], by Chen and Guestrin. The major innovation in XGBoost [18] is that the authors package several optimizations to a Gradient Boosted decision tree algorithm. These optimizations include the use of an approximate algorithm for finding splits in decision trees for sparse datasets, support for caching, and the use of data compression. We use XGBoost in this study because it is a GBDT algorithm, and CatBoost is also a GDBT algorithm. Hence, we are comparing algorithms that adhere to the same general framework.

Although CatBoost is a member of the family of Gradient Boosted Decision Tree algorithms, its implementation is distinct from XGBoost's. The decision trees that CatBoost constructs are balanced, and so tend to be wider than the decision trees XGBoost produces. Prokhorenkova et al. claim this makes CatBoost less prone to overfitting than XGBoost. Furthermore, Prokhorenkova et al. make the case that XGBoost is prone to the phenomenon of target leakage. Target leakage is another form of overfitting that occurs when one uses encoding techniques to deal with high cardinality categorical features such as target encoding [20]. If the learner is given data where certain values of a categorical variable are rare, the learner may tend to overfit if we use target encoding. In this case the algorithm may find a false correlation with rare values of the feature and the label that we encode the feature with. To alleviate target leakage, CatBoost uses ordered boosting. Ordered boosting imposes an order on the samples CatBoost uses to fit constituent decision trees. The ordering of samples ensures that CatBoost does not evaluate a candidate tree with examples it has used to fit the candidate tree, and eventually uses all examples.

Finally, we find many works that evaluate the performance of CatBoost and XGBoost on various machine learning tasks such as [21] and [22]. We also find studies that compare CatBoost to other learners, such as [23] by Zhang et al. To the best of our knowledge, this is the first work that evaluates the performance of CatBoost on extremely imbalanced data, as well as the first study that applies a CatBoost classifier to the task of detecting fraud in Medicare data. In the next section, we give details on how we go about applying CatBoost and XGBoost.

4. Methodology

In our experiments, similar to [17], [7], [16], [5] and [11], we form a dataset from two sources. The first source is data on claims providers submitted to Medicare. This data is furnished by CMS. The second source is a list of healthcare providers that are excluded from submitting claims to Medicare. This list is the LEIE data. The OIG publishes the LEIE data, as mentioned previously. Both OIG and CMS are organizations within the United Sates Federal Government. In this section, we provide details on these two data sources, how we combine them to form datasets, and how we conduct our experiments. Due to space limitations we do not add further details on XGBoost and CatBoost. We recommend interested readers see [18] for more information on XGBoost, and [4] for more information on CatBoost.

4.1. Data Sources

We use Medicare Part B insurance claims data for the years 2012 through 2015. This data originates from a series of publicly available files CMS publishes. We know of these files collectively as Medicare Provider Utilization and Payment Data: Physician and Other Supplier Public Use Files (PUF) [9]. CMS provides a document [24] that describes all the elements of this Medicare Part B insurance claims data. At a high level this data is in character separated value (csv) format, and there is one file for every year. Every record in the data identifies a provider, primarily by the provider's National Provider Identifier (NPI), and secondarily by several elements that give details on the provider's name, demographics and location. Records also contain a provider type that describes the nature of the provider's practice, for example, ophthalmology. For every procedure the provider has submitted a claim to Medicare for, there is one record in the PUF file for the year. This procedure is represented with a Healthcare Common Procedure Coding System (HCPCS) code. In every record, along with the procedure code, there are some aggregate statistics pertinent to it, such as the number of times the provider performed the procedure for the year, and the average amount the provider bills Medicare for. Please see Table 1 for a description of the subset of features of the Medicare data we use for this study. For details on the Medicare Part B data, please see [5].

The second source of data we use in this study is the LEIE data. The OIG updates the LEIE on a monthly basis. It is also a character separated value file. For a complete description of the LEIE please see [10]. In this study, the relevant features of the LEIE are the NPI, the exclusion type, the exclusion date, waiver data and the reinstatement date. In the LEIE csv file, these elements are named: NPI, EXCLTYPE, EXCLDATE, REINDATE, WAIVERDATE, WVRSTATE, respectively.

We derive a label for the Medicare Part B data for a particular year from the LEIE data. If an NPI from the Medicare Part B data appears in the LEIE, and the year for the Medicare Part B data is less than the ending year

TABLE 1. FEATURES USED FOR ALL EXPERIMENTS

Name	Description	Type
provider_type	Medical provider's specialty (or practice)	Categorical *
nppes_provider_gender	Provider gender	Categorical *
nppes_provider_state	The state where the provider is located	Categorical **
hcpcs_code	code used to identify the specific medical service furnished by the provider	Categorical **
line_srvc_cnt	Number of procedures/services the provider performed	Numerical
bene_unique_cnt	Number of distinct Medicare beneficiaries receiving the service	Numerical
bene_day_srvc_cnt	Number of distinct Medicare beneficiary / per day services performed	Numerical
average_submitted _chrg_amt	Average of the charges that the provider submitted for the service	Numerical
average_medicare_ payment_amt	Average amount that Medicare paid after deductible and coinsurance amounts have been deducted for the line item service	Numerical

Single asterisk (*) indicates one-hot encoding is used in both CatBoost and XGBoost; double asterisk (**) indicates feature used only with CatBoost; Feature descriptions from [24]

of the exclusion period, then we label all Medicare Part B data for that year, having that NPI, as fraudulent. We define the ending year of the exclusion period as the exclusion date, plus the mandatory number of years the exclusion type specifies, unless the waiver date, or the reinstatement date come before. The LEIE does not contain data on which procedures in the Medicare Part B data the provider submitted fraudulent claims for, so we label all records submitted prior to the end of the exclusion period as fraudulent. The labeling technique we describe here is the same technique outlined in [6], [16], [17], and [15].

We refer to records labeled as fraudulent as the positive class, and records labeled as non-fraudulent as the negative class. The positive class is very sparse. We follow two different strategies for preparing datasets for XGBoost and CatBoost. In Table 2, we provide positive and negative class counts for the datasets we use as input to CatBoost and XGBoost, as well as the ratios of positive to negative instance counts. Please see Table 2 for class ratio data on the two datasets.

TABLE 2. IMBALANCE RATIOS OF DATASETS

Dataset for …	Positive count	Negative count	Ratio, positive count to negative count
XGBoost	3,691,146	1,409	0.00038
CatBoost	37,224,032	31,411	0.00084

In the following two sections, we provide details on the differences in how we treat the Medicare data for XGBoost versus how we treat it for CatBoost. It is more convenient to work with categorical data with CatBoost. For example, with CatBoost we can use the string value of the HCPCS code as-is, without encoding it. For other learners one must often apply an encoding technique for categorical variables separately before using them.

4.2. Data for XGBoost

We follow the technique outlined in [6] for preparing data for XGBoost. XGBoost cannot use categorical values in their original form, so we use two techniques to convert features of the Medicare Part B data to a numerical form XGBoost can consume. The first technique we use is one-hot encoding, to represent provider gender, and provider type. However, the procedure (HCPCS) code listed in the Medicare Part B data has thousands of possible values. Therefore, one-hot encoding the HCPCS code would be prohibitively expensive in terms of memory consumption, and we do not use HCPCS code directly as a feature. Instead, we aggregate the Medicare Part B data by NPI, year, gender and provider type, and add the minimum, median, maximum, mean, standard deviation, and sum for the features named: line_srvc_cnt, bene_unique_cnt, bene_day_srvc_cnt, average_submitted_chrg_amt, average_medicare_payment_amt. Including these summary statistics for each of these features adds extra information about the procedure the HCPCS code represents, and therefore serves as a proxy for it.

For XGBoost, because we one-hot encode some categorical features, and add summary statistics for numerical features, our dataset has 126 features. The dataset we prepare for XGBoost has a total of 3,692,555 instances.

XGBoost and the dataset we use with it are both similar to the dataset and gradient boosting model used in [7]. In their work, the authors do not use provider state as a feature for their gradient boosted model. Therefore, we also do not include provider state as a feature for XGBoost. We take care to do an experiment with CatBoost where we do not include the provider state as a feature, in the interest of making a fair comparison.

4.3. Data for CatBoost

For CatBoost, we use the same data that we use with XGBoost, but we do not aggregate it by NPI, year, gender, and provider type. We view this as a contribution since it eliminates a pre-processing step authors of related works take, [25], [7], [15] . Therefore, we do not add summary statistics for any of the numerical features like we do for XGBoost. We perform two experiments with CatBoost. For both, we retain the HCPCS code, and for one we use the provider state feature as well. Therefore, our experiments with CatBoost involve datasets with 7 or 8 features. For XGBoost, we did not retain the HCPCS code, or provider state. For this initial work, one of our goals was to use a current GBDT algorithm with the same dataset as used in previous research to establish a baseline of performance in

terms of AUC; hence, our choice not to include HCPCS code or provider state in the experiment we do with XGBoost. Please recall that we include summary statistics of other features related to HCPCS code in an attempt to retain some information inherent in the HCPCS code. An opportunity for future work would be to experiment with different embedding techniques for the HCPCS code and provider state with XGBoost.

CatBoost automatically handles categorical features without our needing to pre-process data. Hence, we are able to easily use the HCPCS code, provider gender, provider type and provider state features of the Medicare Part B data as features for CatBoost. As a result, we are able to use a dataset having fewer features with CatBoost than with XGBoost. This does not necessarily skirt the curse of dimensionality [26]. CatBoost expands the dimensionality of its input categorical features beneath the level of its application programming interface. Internally, CatBoost uses one-hot encoding for categorical features with small cardinality. We use the default setting of 255 for the maximum size of the set of distinct values that CatBoost will use one-hot encoding for. So, only the HCPCS code feature is not one-hot encoded in our experiments. We use the default method for the encoding technique CatBoost uses in classification tasks, for categorical variables with a set of distinct values larger than the minimum size for one-hot encoding. Please see the Section titled "Transforming categorical features to numerical features in classification" in [27] for a detailed discussion on this encoding technique. For a detailed review of techniques for encoding categorical variables in general, please see Hancock and Khoshgoftaar [28]. There is an opportunity for future research to investigate whether we can find a better value for the threshold value CatBoost uses to decide whether to use one-hot encoding, since the threshold is configurable. CatBoost can also use different encoding techniques, for high cardinality categorical variables. The encoding technique is determined by parameters the user specifies. Also, the encoding techniques available depend on the type of machine learning task — classification, regression, or multi-class classification — and the method of transforming categorical features the user specifies when setting CatBoost hyper-parameters. Further details of the various embedding techniques CatBoost uses are outside the scope of this study. Our goal is to establish a relative baseline comparison of the performance of XGBoost and CatBoost. Hence, we are only interested in using hyper-parameter settings for both algorithms as close to their default values as possible. Interested readers should see the documentation in [29].

4.4. Data Sampling

Previous studies generally indicate that the best technique for addressing class imbalance for the task of fraud detection in Medicare data is random undersampling. Therefore, we use random undersampling as well. However, the best ratio to balance classes is somewhat less clear. On the one hand, we have [7] that shows the best results with a 80% positive to 20% negative class ratio. On the other hand, we have [25] that reports the best performance with a balanced 50:50 ratio. Since [25] reports performance of a gradient boosted decision tree algorithm with a 50:50 ratio we use the same ratio here. It should be noted, however, that applying random undersampling to this class ratio reduces the size of datasets and risks data loss of the negative class. We repeat experiments ten times to mitigate that risk. We see an opportunity for future research to investigate the impact of varying the class ratio on the performance of CatBoost. We use stratified 5-fold cross-validation to train and score XGBoost and CatBoost. We apply random undersampling to the data we use in fitting XGBoost and CatBoost, but when we calculate the AUC value of the fitted classifiers' outputs, we do not apply random undersampling to the input data that we feed to the fitted classifiers. This concludes our discussion on data we use in this study; now we move on to cover the algorithms we apply the data to.

4.5. Models

In order to make a fair baseline comparison, we decided to use hyper-parameters as close to default values as possible for both XGBoost and CatBoost. In this study we do not compare the running times of CatBoost and XGBoost. We reserve such a comparison for future work. We do not attempt any hyper-parameter tuning. The hyper-parameters we use for XGBoost are listed in Table 3. The hyper-parameters we use for CatBoost are listed in Table 4.

TABLE 3. XGBOOST HYPER-PARAMETERS

Hyper-parameter	Value
tree_method	gpu_hist
objective	binary:logistic
learning_rate	0.1
max_depth	6

For XGBoost and CatBoost, we specify that the algorithms use GPU's because we found using GPU's speeds up the model training process. For XGBoost, we found that, in order for it to train in a reasonable amount of time, we needed to set the learning rate explicitly to 0.1 and the maximum depth of its constituent decision trees to 6. As a check on label data quality, we set the XGBoost classifier's objective function to binary-logistic. We use this setting to force an error if there are more than two possible values for labels. The only hyper-parameter settings we use for CatBoost are those that instruct CatBoost to use GPU's, and one that instructs CatBoost on which features are categorical.

We use the Python application programming interfaces (API's) for both XGBoost and CatBoost. The Python version we use is 3.7.3. We use one GPU to fit both CatBoost and XGBoost models. We use Python scikit-learn [30] for stratified 5-fold cross validation and AUC calculation.

TABLE 4. CATBOOST HYPER-PARAMETERS

Hyper-parameter	Value
cat_features	provider_type nppes_provider_gender nppes_provider_state
task_type	GPU
devices	0

5. Results and Discussion

In this section, we present results detailing XGBoost and CatBoost's performance in the task of identifying claims data labeled as fraudulent in the Medicare Part B dataset. We report performance in terms of AUC.

We perform ten iterations of stratified 5-fold cross validation for XGBoost and CatBoost using the data we describe in section 4.1. For all experiments, we use random undersampling to balance positive and negative classes to a one-to-one ratio. The AUC values in Table 5 are the mean of a total of 50 AUC values that we record for each of the five folds and ten iterations of each classifier. We do not use provider state as a feature for XGBoost for reasons we explain in Section 4.1. Largely because with XGBoost, the computationally expensive pre-processing step of one-hot encoding is required for categorical features, such as provider state. In contrast, with CatBoost we perform experiments with and without including the provider state categorical feature in the data set. We show a significant increase in the AUC value when we include provider state as a feature for CatBoost. We report results for CatBoost without provider state in the row labeled "CatBoost-HCPCS" and results for CatBoost with provider state in the row labeled "CatBoost-HCPCS-state" in Table 5.

TABLE 5. CATBOOST AND XGBOOST AUC AND STANDARD DEVIATION

Experiment	Mean AUC	SD AUC
XGBoost	0.76154	0.01179
CatBoost-HCPCS	0.78511	0.00221
CatBoost-HCPCS-state	0.89015	0.00182

In our experiments, CatBoost outperforms XGBoost by a mean AUC difference of 0.02. When we add the provider state as a feature for CatBoost, we see that CatBoost increases its mean AUC lead to 0.12. This supports Branting et al.'s conclusion from their ablation study, that geospatial features provide Medicare fraud detection models important information [8]. In the next section, we confirm that the results are statistically significant.

5.1. Statistical Analysis

In order to be sure that the differences in mean AUC that we record from our experiments are statistically significant, we employ a z-test to estimate the probability that mean AUC values for the XGBoost and CatBoost-HCPCS classifiers are different, and another z-test to estimate the probability that the mean AUC values of the CatBoost-HCPCS and CatBoost-HCPCS-state experiments are different. The p-values for both z-tests are 0. Therefore, we conclude at a 99% confidence level that the mean AUC values of the XGBoost and CatBoost-HCPCS experiments, and the mean AUC values of the CatBoost-HCPCS and CatBoost-HCPCS-state experiments are different in a statistically significant sense. Thus, we can state CatBoost performs significantly better than XGBoost for Medicare fraud detection.

6. Conclusion

In this study, we investigate XGBoost and CatBoost's ability to classify an extremely imbalanced dataset of Medicare claims data, in the task of Medicare fraud detection. We do not intend to show CatBoost is superior to XGBoost in this classification task. Rather, we establish a baseline performance using existing methods, and show that CatBoost attains performance similar to the baseline. We show that CatBoost's default built-in handling of categorical features is effective, even for high-cardinality features such as the HCPCS code in Medicare claims data. We take advantage of this to eliminate pre-processing steps authors of some related works take. Specifically, we show CatBoost obtains a mean AUC value of 0.78511, which is higher than the mean AUC of 0.76154, that we obtain with XGBoost. Furthermore, when we include the provider state feature in the Medicare data as input for CatBoost, we obtain a mean AUC value of 0.89015. Both performance improvements are statistically significant at a 99% confidence interval (with p-value 0). The significant improvement in AUC for CatBoost when the provider state feature is included is in alignment with previous research [8] that suggests location data is an important feature for Medicare fraud detection.

Our research indicates one may rely on CatBoost's internal mechanisms for representing high cardinality categorical variables. In this study, adding a categorical feature improves CatBoost's AUC score. Moreover, since the classification task is to identify potential Medicare fraud, our results have a practical value as a contribution to research on machine leaning for Medicare fraud detection. We have set the stage for future experiments to investigate if performance in terms of AUC will improve when we tune the hyper-parameters of XGBoost and CatBoost, use more categorical features. We find CatBoost to be a viable tool for Medicare fraud detection.

Acknowledgment

The authors would like to express their gratitude to the reviewers at the Data Mining and Machine Learning Laboratory of Florida Atlantic University for the help in preparing this study.

References

[1] Centers for Medicare & Medicaid Services. Get started with medicare. [Online]. Available: https://www.medicare.gov/sign-up-change-plans/get-started-with-medicare

[2] Centers For Medicare & Medicaid Services. (2018) Trustees report & trust funds. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/ReportsTrustFunds/index.html

[3] S. J. Bagdoyan. (2018) Testimony before the subcommittee on oversight, committee on ways and means, house of representatives. [Online]. Available: https://www.gao.gov/assets/700/693156.pdf

[4] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 6638–6648. [Online]. Available: http://papers.nips.cc/paper/7898-catboost-unbiased-boosting-with-categorical-features.pdf

[5] R. A. Bauder and T. M. Khoshgoftaar, "A novel method for fraudulent medicare claims detection from expected payment deviations (application paper)," in *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, July 2016, pp. 11–19.

[6] R. A. Bauder, R. da Rosa, and T. M. Khoshgoftaar, "Identifying medicare provider fraud with unsupervised machine learning," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, July 2018, pp. 285–292.

[7] R. A. Bauder and T. M. Khoshgoftaar, "Medicare fraud detection using machine learning methods," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2017, pp. 858–865.

[8] L. K. Branting, F. Reeder, J. Gold, and T. Champney, "Graph analytics for healthcare fraud risk estimation," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug 2016, pp. 845–851.

[9] (2019) Medicare provider utilization and payment data: Physician and other supplier. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Physician-and-Other-Supplier

[10] LEIE. (2017) Office of inspector general leie downloadable databases. Acessed: 2020-05-17. [Online]. Available: https://oig.hhs.gov/exclusions/index.asp

[11] R. A. Bauder and T. M. Khoshgoftaar, "A probabilistic programming approach for outlier detection in healthcare claims," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2016, pp. 347–354.

[12] J. Weaver and D. Chang. South florida ophthalmologist emerges as medicare's top-paid physician. [Online]. Available: http://www.miamiherald.com/news/local/community/miami-dade/article1962581.html

[13] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "Evaluation measures for models assessment over imbalanced data sets," *Journal Of Information Engineering and Applications*, vol. 3, no. 10, 2013.

[14] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions in: Proc of the 3rd international conference on knowledge discovery and data mining," 1997.

[15] R. A. Bauder and T. M. Khoshgoftaar, "Medicare fraud detection using random forest with class imbalanced big data," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, July 2018, pp. 80–87.

[16] J. M. Johnson and T. M. Khoshgoftaar, "Deep learning and data sampling with imbalanced big data," in *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, July 2019, pp. 175–183.

[17] J. M. Johnson and T. M. Khoshgoftaar, "Deep learning and thresholding with class-imbalanced big data," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec 2019, pp. 755–762.

[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[19] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367 – 378, 2002, nonlinear Methods and Data Mining. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167947301000652

[20] D. Micci-Barreca, "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems," *SIGKDD Explor. Newsl.*, vol. 3, no. 1, p. 27–32, Jul. 2001.

[21] G. Huang, L. Wu, X. Ma, W. Zhang, J. Fan, X. Yu, W. Zeng, and H. Zhou, "Evaluation of catboost method for prediction of reference evapotranspiration in humid regions," *Journal of Hydrology*, vol. 574, pp. 1029 – 1041, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022169419304251

[22] E. A. Daoud, "Comparison between xgboost, lightgbm and catboost using a home credit dataset," *International Journal of Computer and Information Engineering*, vol. 13, no. 1, pp. 6 – 10, 2019. [Online]. Available: https://publications.waset.org/vol/145

[23] H. Zhang, R. Zeng, L. Chen, and S. Zhang, "Research on personal credit scoring model based on multi-source data," *Journal of Physics: Conference Series*, vol. 1437, p. 012053, Jan 2020.

[24] CMS Office of Enterprise Data and Analytics, "Medicare fee-for-service provider utilization & payment data physician and other supplier," 2017. [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/Medicare-Physician-and-Other-Supplier-PUF-Methodology.pdf

[25] M. Herland, R. A. Bauder, and T. M. Khoshgoftaar, "The effects of class rarity on the evaluation of supervised healthcare fraud detection models." *Journal of Big Data*, vol. 6, no. 1, p. 1, 2019.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, July 2016.

[27] (2020) Transforming categorical features to numerical features. [Online]. Available: https://catboost.ai/docs/concepts/algorithm-main-stages_cat-to-numberic.html#algorithm-main-stages_cat-to-numberic

[28] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks." *Journal of Big Data*, vol. 7, no. 1, pp. 1 – 41, 2020.

[29] (2020) Python package training parameters. [Online]. Available: https://catboost.ai/docs/concepts/python-reference_parameters-list.html#python-reference_parameters-list

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.