A Big Data Analytics Framework for Forecasting Rare Customer Complaints

A use case of predicting MA members' complaints to CMS

Donghui Wu
PCCI
Dallas, TX, USA
donghui.wu@ieee.org

***Abstract*— Centers for Medicare & Medicaid Services (CMS) publishes Medicare Part C Star Ratings each year to measure the quality of care of Medicare Advantage (MA) contracts. One of the key measures is Complaints about the Health Plan, which is captured in Complaints Tracking Module (CTM). Complaints resulted in CTM are rare events: for MA contracts with 2-5 star ratings, number of complaints for every 1,000 members range from .10 to 1.84 over last 5 years. Reducing number of complaints is extremely important to MA plans as they impact CMS reimbursements to MA plans. Forecasting and reducing complaints is an extremely technically challenging task, and involves ethics considerations in patients' rights and privacy. In this research, we constructed a big data analytics framework for forecasting rare customer complaints. First, we built a big data ingestion pipelines on a Hadoop platform: a) Ingest MA plan's customer complaints data from CTM from past 3 years. b) Ingest health plan's call center data for MA members from past 3 years, including both structured data and unstructured text script for the calls. c) Ingest MA members' medical claims, including members' demographics and enrollment history. d) Ingest MA members' pharmacy claims. e) Integrate and unified data from above sources, and enrich the data with additional engineered features into a big wide table, one row per member for analysis and modeling. Second, we designed a unique decision tree based Large Ensemble with Over-Sampling (LEOS) algorithm, which mimics random forest but with extreme oversampling of target class to increase bias, and leverages the parallel computing of Hadoop clusters by generating thousands of fixed size training data sets, and for each such dataset training a decision trees with similar fixed tree structure, and ensemble them. Third, we validated our framework and LEOS learning algorithm with real data, and also discussed ethics issues we encountered in handling data and applying findings from research.**

Keywords-CMS Star Ratings; customer complaints; big data analytics; ensemble methods; ethics; call center data; medical claims

I. INTRODUCTION

One of the Centers for Medicare & Medicaid Services' (CMS) most important strategic goals is to improve the quality of care and general health status for Medicare beneficiaries. CMS publishes the Part C and D Star Ratings each year to: measure quality in Medicare Advantage (MA) and Prescription Drug Plans (PDPs or Part D plans), assist beneficiaries in finding the best plan for them, and determine MA Quality Bonus Payments [1]. The Star Ratings are displayed in Medicare Plan Finder [2], so Medicare beneficiaries can consider both quality and cost when selecting a plan for enrollment. The Star Ratings have additional marketing and financial impacts for health plans. The 5-star rating health plans can market year-round. Beneficiaries can join these plans at any time via a special enrollment period (SEP). Medicare Plan Finder disables online enrollment of consistently low performing plans (2-stars or lower). In addition, Affordable Care Act established CMS' Star Ratings as the basis of Quality Bonus Payments to MA plans [3]. Therefore, MA plans are motivated to improve their star ratings.

Medicare Advantage with prescription drug coverage (MA-PD) contracts are rated on up to 44 unique quality and performance measures; MA-only contracts (without prescription drug coverage) are rated on up to 32 measures; and stand-alone PDP contracts are rated on up to 15 measures. Each year, CMS conducts a comprehensive review of the measures that make up the Star Ratings, considering the reliability of the measures, clinical recommendations, feedback received from stakeholders, and data issues. The Star Ratings measures span 5 broad categories: 1) Outcomes; 2) Intermediate Outcomes; 3) Patient Experience; 4) Access; 5) Process [4]. Figure 1 shows the statistics on MA customer complaints.

This paper is focused on improving Patient Experience scores to boost MA plans star ratings. One of the key measure of patient experience is number of complaints about the health per thousand members. Our approach is to predict the members who are more likely to file a complaint and then proactively work with these at-risk members to resolve their issues before complaints are filed with CMS.

The paper is organized as following: Section I, Introduction; Section II describes the big data analytics framework used in this paper; including the data ingestion, integration and enrichment process. Section III describes the setup of predictive modeling problem and the details of LEOS algorithm for predicting rare customer complaints, a decision tree based large ensemble method with over-sampling of target class. The implementation and experiment results are provided in Section IV. Finally in Section V we discuss some ethic issues encountered in analyzing the problem, design and implementing the solutions.

II. THE BIG DATA ANALYTICS FRAMEWORK

The big data analytics framework is designed for a Hadoop based cluster environment, which includes HDFS, HIVE, Pig, Yarn, and other components. The model training is done using R with MapReduce and Parallelization packages.

A. Ingest the Datasets

First, ingest MA plan's complaints data from CMS CMT system from year 2012-2014; then ingest MA plan's call center data from 2012-2014, including structured data, e.g. caller, call time, duration of call, call category, and unstructured data, brief notes from each call; third, ingest plan's membership enrollment history, medical claims data, and pharmacy claims data from 2012-2014. These datasets formed a data lake on Hadoop system.

B. Integrate and enrich the data sets

The ingested data sets are combined, linked and integrated into a big wide table, one row per member for easy analysis. The big wide table are further enriched to create hundreds of additional member level predictors based on raw data, e.g. number of calls leading up to a complaint or end of experience period, interval between call; comorbidity and severity of conditions and indication for future risk from medical and pharmacy claims data, etc.

III. THE LEOS ALGORITHM

A. Analysis of the Problem

Complaints about health plan resulted in CMS CTM are rare events: for MA contracts with 2-5 star ratings, number of complaints for every 1,000 members range from .10 to 1.84 from year 2011 – 2014 as shown in Figure 1. Forecasting and reducing complaints from MA members is an extremely technically challenging task because these are rare events and are very hard to predict. We set up the problem as using past year data to predict next quarter's member complaints. It's a really difficult task of learning a classification model from extremely imbalanced classes.

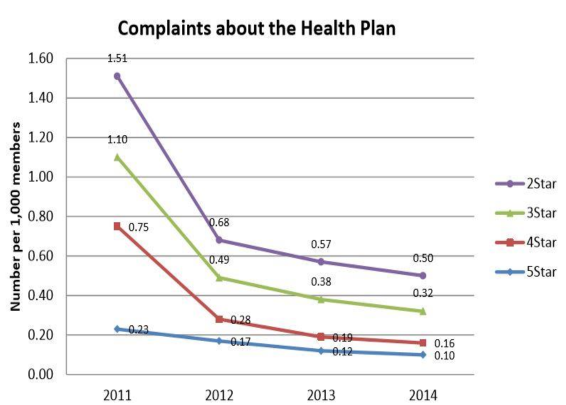

Figure 1. Complaints about the health plans, source: CMS 2014 plan star ratings and measures.

B. Description of LEOS algorithm

Ensemble methods like AdaBoost [5], bagging [6,7], and random forest [8,9] have been extremely popular in practice for its overall good performance and tolerance of noises in data. For our two class classification problem, the target class, members who files complaint(s) with CMS, is extremely rare as shown in Figure 1. It's a learning task with extremely imbalanced data set.

We designed a unique decision tree based Large Ensemble with Over-Sampling (LEOS) algorithm, which mimics random forest algorithm but with extreme oversampling of target class to increase bias towards target class. The algorithm also leverages the parallel computing of Hadoop clusters by generating thousands of training data subsets with predetermined size and pseudo prevalence rate much higher than actual, and for each such subset a decision tree model is built from a randomly selected subset of features available. The final LEOS model is the ensemble of these thousands of decision tree models.

With limited number of training samples with target label, and extremely low prevalence rate, e.g. 1% or lower, none of the existing algorithms can produce a model with reasonable sensitivity and positive predictive value. The learning task is just too difficult. The idea behind the LEOS algorithm is the following: First, over-sampling of target class is needed to train a model with reasonable training accuracy. Even though it is based on a biased distribution different from original distributions, the leading predictors and relationship that discriminate targets and non-targets are still meaningful. Second, a large number of models are produced to form an ensemble: each model is based on a randomly selected over-sampled small subset of training data with a randomly selected subset of features. Third, the test dataset is scored by the model ensemble as final predictions.

Given an extremely imbalanced training data set S(X, Y), split it into $T_1(X, 1)$ as subset of targeted class and $T_0(X, 0)$ as subset of non-targeted class. Denotes the number of samples in S as N(S), the number of samples in T_1 as $N(T_1)$ and the number of samples in T_0 as $N(T_0)$, then $N(S) = N(T_0) + N(T_1)$ and $N(T_0) >> N(T_1)$. For a given size $n << N(S)$ and pseudo prevalence rate p ($10\% =< p <= 40\%$), threshold D_n (e.g.1,000), the following is the outline of LEOS algorithm:

1) Create a randomly selected training data subset with over-sampling of target class

*a) Randomly selected p*n target samples from target class T_1 with replacement;*

*b) Randomly selected (1-p)*n non-target samples from non-target class T_0 without replacement;*

c) Combine the two to form a training subset t with n training samples and a pseudo prevelance rate of p.

d) Keep a randomly selected subset of features in training subset t and discard the rest features, to form a shrinked training subset t.*

2) Train a decision tree model d from a given training subset t*.*

3) Record and compute the importance (w_i) of predictor x_i based on the location of splits within the decision tree d.*

e) If predictor x_i is chosen as a split at a node with depth l, the importance w_i += 1/l, at root, l=1.

4) Repeat step 1) – 3) until $T_0(X, 0)$ is exhausted.

5) If number of desison trees is smaller than threshold D_n, Repeat Step 1) - 4).

6) Output LEOS model as the ensemble of all decision trees.

7) Output the importance of predictor (x_i) as the sum of (w_i) across all decision trees.

At the end of LEOS algorithm, we obtained an ensemble of large number of decision trees, each trained from a randomly selected small subset of training data with over sampling of target class. We also obtained importance (w_i) of each predictor (x_i) in the LEOS model. Simply ranking predictors (x_i) according to (w_i), produces leading predictors.

IV. Implementation and Experiments

The implementation LEOS is done in R with MapReduce and parallelization packages. In implementation, the complexity of LEOS algorithm is encapsulated in the training data subset generator. It generates training data subsets according to given parameters: training set S(X,Y), subset size n, pseudo prevalence rate *p*, and threshold D_n. A decision tree d_i is trained for each generated training data subset s_i. The importance vector (w_i) is also created for all predictors (x_i) for each decision tree d_i. The final LESO model is the ensemble of all decision trees (d_i) and importance vector (w_i).

The LEOS algorithm is applied to a MA plan with over 750,000 members, with about .2% members filed complaints with CMS in year 2014 and about .05% members filed complaints each quarter. 1/3 of the target class and 1/3 non-target class are reserved as test data set, the other 2/3 are used for training. A large number of experiments are conducted with a grid search of parameters shown in table 1.

n	**p**	**D_n**
100	15%	5000
200	**20%**	2500
500	25%	**1000**
1000	30%	500

Table 1. The parameters for grid search

The experiments showed that with training subset size of 500, pseudo prevalence rate 20% and minimal ensemble size of 1000 decision trees, LEOS produced somewhat acceptable test results for quarterly predictions with about 33% sensitivity and 30% PPV, which is about 600 times more accurate than random selection.

V. Discussion and Ethics Considerations

In exploring the patients' data and applying the findings from learning, we encountered numerous ethics related issues and impacted our choices of implementation, and approaches of addressing the issues.

The initial business goal is very clear: reduce number of complaints filed with CMS with advanced analytics. However, the more natural approach seems to improve product offerings, improve customer services, and resolve customer issues for all customers. Instead, the approach taken is that predicting those customers who are more likely to file complaints and address their issues more effectively. In effect, the plan ignores those customers who are not happy with their services but unable to or unwilling to file complaints with CMS.

In addition, from analyzing phone records and complaints, a large number of complaints are about enrollment and benefit eligibility, which in most cases are not necessary the faults of insurance plan. Resolving these issues may reduce the number of complaints, but may not necessary in line of policies or fair to other members. Sometimes, the model reveal issues of the members unrelated to complaints, for example high risk for certain conditions, should we ignore risks, but focus on identifying member complaints? There are numerous ethics related issues should be properly addressed in line of business goals, corporate missions, and personal morale.

References

[1] CMS Star Ratings: 2017 Star Rating, https://www.cms.gov/Newsroom/MediaReleaseDatabase/Fact-sheets/2016-Fact-sheets-items/2016-10-12.html, 11/14/17.

[2] Medicare Plan Finder, https://www.medicare.gov/find-a-plan/questions/home.aspx, last accessed 11/14/17.

[3] MEDICARE ADVANTAGE PROGRAM PAYMENT SYSTEM, http://www.medpac.gov/docs/default-source/payment-basics/medpac_payment_basics_16_ma_final.pdf, last accessed 11/14/17.

[4] Part C and D Performance Data, https://www.cms.gov/Medicare/Prescription-Drug-Coverage/PrescriptionDrugCovGenIn/PerformanceData.html, last accessed 11/14/17.

[5] Freund, Yoav; Schapire, Robert E. "A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences. 55: 119 – 139, August 1997.

[6] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and Regression Trees., Chapman & Hall, New York, 1984.

[7] L. Breiman. Bagging predictors. Machine Learning, 24:123–140, 1996.

[8] L. Breiman. Random forests. Machine Learning, 45:5–32, 2001.

[9] Gerard Biau, "Analysis of a Random Forests Model", Journal of Machine Learning Research 13, pp. 1063-1095, 2012.