Heritage Provider Network Health Prize Round 3 Milestone: Team crescendo's Solution

Rie Johnson Tong Zhang

1 Introduction

This document describes our entry nominated for the second prize of the Heritage Network Provider Health Prize Round 3 Milestone as required by the rules. The entry is a weighted sum (called 'blend' in this document) of multiple runs, each of which was generated by one of the following methods:

- *Regularized greedy forest* (RGF) [6].
- *Gradient boosting decision tree* (GBDT) [5].
- Linear models with L_2 regularization, trained for the residual of either RGF or GBDT predictions. Some linear model runs used features generated by *Alternating structure optimization* (ASO) [1].
- *Random forests* [3] trained for the residual of either RGF or GBDT predictions.

The difference among the runs produced by the same algorithm is the features. Among these, RGF is a method to learn tree ensembles, like GBDT. Our original motivation to enter the competition was to test RGF in a competitive setting. In our experiments, RGF consistently produced more accurate models than GBDT, but blending RGF and GBDT runs produced even more accurate models.

2 Algorithms

All the models were trained for minimizing square error with the log-scale target ($\log(x + 1)$). Corresponding references should be consulted for the content of the algorithms. This section describes how we used the algorithms. More detailed information required for replicating the results will be given in the Appendix.

2.1 Regularized greedy forest (RGF)

Implementation of RGF is available at `http://riejohnson.com/rgf_download.html`, which is open software issued under GNU Public License V3. As we performed regularization on leaf-only models with the extension (described in [6]), there were two L_2 regularization parameters to be set: one for weight optimization and the other for tree learning. All the RGF runs set these parameters to 0.5 and 0.005, respectively. The model size in terms of the number of leaf nodes in the forest (tree ensemble) was set to 20000 unless otherwise specified. The other parameters were set to the default of the system.

2.2 Gradient boosting decision tree (GBDT)

A well-known implementation of GBDT is the R package `gbm` [7] though we used our own implementation for convenience.

The shrinkage parameter, tree size (in terms of the number of leaf nodes), and the minimum training data points per node were set to 0.01, 20, and 100, respectively. We did not perform data sampling. The model size in terms of the number of leaf nodes in the forest was set to 20000 unless otherwise specified.

2.3 Random forests combined with RGF/GBDT

We trained random forests for the residual of either an RGF run or a GBDT run. That is, the training target for the i-th data point was set to $y_i - f(\mathbf{x}_i)$ where y_i is the original target (DaysInHospital in the log-scale) and $f(\mathbf{x}_i)$ is the prediction made by an RGF (or GBDT) model, which serves as *base prediction*. The prediction on test data was done by adding to the base prediction the residual prediction made by a random forest model. Parameters used for random forest training are described in the Table 8 in the Appendix.

2.4 Linear models combined with RGF/GBDT and ASO

The linear models were trained for the residual of either an RGF run or a GBDT run. More precisely, for the i-th data point, let $\mathbf{x}_i$ be the feature vector and let $\tilde{y}_i$ be the residual of the base prediction (i.e., $\tilde{y}_i = y_i - f(\mathbf{x}_i)$). Then the training objective of the linear models was to obtain weight vector $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left[\frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top \mathbf{x}_i - \tilde{y}_i)^2 + \lambda \mathbf{w}^\top \mathbf{w} \right]$. The L_2 regularization parameter λ was set to 0.01.

In some runs, we used Alternating structure optimization (ASO) [1] to generate additional features. Essentially, ASO learns new features through dimensionality reduction of predictors that are trained for *auxiliary tasks*. At a high level, the idea underlying ASO is *multi-view learning* as analyzed in [2]. The auxiliary tasks are in the form of predicting one *view* of the features (*target view*) based on another view (*feature view*); for example, to predict how many claims with PlaceSvc=Ambulance the member has, using the Vendor information as only input (i.e., zeroing out other features). The ASO procedure we performed was as follows.

1. Define m auxiliary tasks.
2. Train linear predictors for the auxiliary tasks, which results in m weight vectors, and let $\mathbf{W}$ be a matrix whose columns are the m weight vectors.
3. Compute $\mathbf{W}$'s singular value decomposition, and let $\mathbf{U}_k$ be a matrix whose columns are the left singular vectors of $\mathbf{W}$ corresponding to the k largest singular values.
4. Compute $\mathbf{Z} = \mathbf{U}_k^\top \mathbf{X}$ where $\mathbf{X}$ is the original feature matrix whose columns are feature vectors. Then the columns of $\mathbf{Z}$ are the new additional feature vectors of k dimensions.

The auxiliary tasks we defined will be given in Appendix A.3.

3 Features

Generation of individual features mostly follows or extends [4]. However, notable differences are that we did not do any elaborate feature selection, and that we combined the extracted features quite differently – most of our runs used the features generated by aggregating claims in a one-year period and a two-year period simultaneously. Details will be given in Appendix A.

4 Blending and post processing

4.1 Blending to produce the winning submission

The nominated entry was a blend of five runs, and their public Leaderboard scores and blend weights are shown below.

name	public score	weight
run#1	0.457284	0.707974
run#2	0.458169	0.381684
run#3	0.458471	0.215296
run#4	0.458643	0.269279
run#5	0.459501	−0.574206

The blend weights were determined by a slight variation (described later) of Section 7.1 of [8], which approximately solves ridge regression on the test data using the Leaderboard scores. As we understand it, this blending method was also used by the previous milestone winners. Although [8] should be consulted for details, it is worth mentioning that the center piece of this approximation is the fact that in the solution to ridge regression $(\mathbf{X}^\top\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$ on the n test data points, the only unknown term $\mathbf{X}^\top\mathbf{y}$ (because the target $\mathbf{y}$ is unknown) can be closely approximated using the Leaderboard scores. That is, if we let $\mathbf{p}^{(j)}$ be the j-th run (predictions), then the j-th component of vector $\mathbf{X}^\top\mathbf{y}$ is $\mathbf{p}^{(j)} \cdot \mathbf{y}$, and we have:

$$\mathbf{p}^{(j)} \cdot \mathbf{y} = -\frac{1}{2}\left(\|\mathbf{p}^{(j)} - \mathbf{y}\|^2 - \|\mathbf{p}^{(j)}\|^2 - \|\mathbf{y}\|^2\right) .$$

$\|\mathbf{p}^{(j)} - \mathbf{y}\|^2$ can be approximated by $n \cdot s_j$ where s_j is the Leaderboard score of the j-th run $\mathbf{p}^{(j)}$, and $\|\mathbf{y}\|^2$ can be approximated using the all-zero benchmark provided by the organizer.

Our "slight variation" was to train for the residual, using the average of the five runs $\mathbf{b} = \frac{1}{5}\sum_{j=1}^{5}\mathbf{p}^{(j)}$ as *base prediction* (introduced in Section 2.3), and to use $\mathbf{p}^{(j)} - \mathbf{b}$ in place of $\mathbf{p}^{(j)}$. In this variation the j-th component of the unknown term is $(\mathbf{p}^{(j)} - \mathbf{b}) \cdot (\mathbf{y} - \mathbf{b})$ instead of $\mathbf{p}^{(j)} \cdot \mathbf{y}$ and its approximation simply reduces to the approximation of $\mathbf{p}^{(i)} \cdot \mathbf{y}$ ($i = 1, \ldots, 5$). The regularization parameter λ was set to 0.0001.

4.2 Blending to produce the five runs

Each of the five runs was a blend of multiple runs. This blending was done by ridge regression trained on the training data using 5-fold cross validation results. In this blending, we again used the average of the input runs as base prediction and trained for residual. The features were the differences of each run $\mathbf{p}$ from the base prediction $\mathbf{b}$ and three vectors per run given by: $\mathbf{v}_{1,i} = \begin{cases} \mathbf{p}_i - \mathbf{b}_i & \text{if } \mathbf{p}_i < 0.5 \\ 0 & \text{otherwise} \end{cases}$, $\mathbf{v}_{2,i} = \begin{cases} \mathbf{p}_i - \mathbf{b}_i & \text{if } \mathbf{p}_i \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$, and $\mathbf{v}_{3,i} = \begin{cases} \mathbf{p}_i - \mathbf{b}_i & \text{if } \mathbf{p}_i \geq 1 \\ 0 & \text{otherwise} \end{cases}$, where subscript i indicates the i-th vector component. The regularization parameter was set to 0.0005 for Run#2, 0.0001 for Run#5, and 0.001 for the rest.

The individual runs blended into five runs will be described in Appendix C .

4.3 Post processing

Post processing was applied to the final blend. Let S_a be the set of test data points for which Age is missing. Its complement $\bar{S}_a$ is the set of test data points for which Age is given. Similarly, define S_s and $\bar{S}_s$ for Sex. Our post processing was to match the average of the predictions within these four sets to the *true average* – first within S_s and $\bar{S}_s$ and then within S_a and $\bar{S}_a$. The "true average" of the targets within a subset can be estimated from the Leaderboard scores by noting the following. Let s be the Leaderboard score of a "constant model" in [4], which sets a constant for the data points in S and 0 for the data points in $\bar{S}$. Then we have:

$$n \cdot s^2 \approx \sum_{i \in S}(c - y_i)^2 + \sum_{i \in \bar{S}} y_i^2 = |S|c^2 - 2c\sum_{i \in S} y_i + \sum_{i=1}^{n} y_i^2 .$$

where c is the constant in log-scale. On the right-hand side, $\sum_{i \in S} y_i$ in the second term is the desired quantity, the third term can be approximated using the all-zero benchmark, and the first term is known.

Finally, we convert the log-scale prediction values to the desired scale and truncate the results into $[0, 15]$.

References

[1] Rie Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[2] Rie Ando and Tong Zhang. Two-view feature generation model for semi-supervised learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.

[3] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[4] Phil Brierley, David Vogel, and Randy Axelrod. Heritage Provider Network Health Prize Round 1 Milestone Prize: How we did it – Team 'Market Makers'. https://www.heritagehealthprize.com/c/hhp/leaderboard/milestone1, 2011.

[5] Jerome Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29, 2001.

[6] Rie Johnson and Tong Zhang. Learning nonlinear functions using regularized greedy forest. Technical report, arXiv:1109.0887v5, 2012.

[7] Greg Ridgeway. Package 'gbm'. http://cran.r-project.org/web/packages/gbm/gbm.pdf, 2012.

[8] Andreas Töscher and Michael Jahrer. The BigChaos solution to the Netflix Grand Prize. http://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf, 2009.

A Features

We first define several feature sets, which will be combined to compose ‘datasets’ in the next section. See [4] for the definition of “velocity”, “range”, and “AdmissionRisk”; and conversion of categorical values to numerical values such as DSFS and LengthOfStay.

A.1 Features derived from the member information: m1

Feature set: m1

Age	numeric
Age missing	binary flag
Age.0-9	binary flag
Age.10-19	binary flag
Age.20-29	binary flag
Age.30-39	binary flag
Age.40-49	binary flag
Age.50-59	binary flag
Age.60-69	binary flag
Age.70-79	binary flag
Age.80-	binary flag
Male	binary flag
Female	binary flag
Gender unknown	binary flag
ClaimsTruncated	binary flag

A.2 Features derived from Claim data

Notation

- ‘1y’: aggregate the claim data in a one-year period.
- ‘2y’: aggregate the claim data in a two-year period.
- *count*: count of claims for that member in which the corresponding category appears. Blank (missing value) was treated as a category only for Vendor, ProviderID, and PCP, and ignored (i.e., not counted) for others.
- The numbers in the parentheses are the number of features corresponding to the description. For example, “PlaceSvc count (8)” in the first line means that there are eight features each of which is the count of one of eight PlaceSvc categories (‘Ambulance’, ‘Home’, and so on) for the member. If omitted, the number of features is one.
- *ratio*: count divided by the number of claims for the member. Blank (missing value) was treated as a category.

A.2.1 Feature sets t1, t3, w1, w2, and w3

Description	Type	Feature set				
		t1	t3	w1	w2	w3
PlaceSvc count (8)	numeric	1y	1y	2y	2y	2y
Specialty count (12)	numeric	1y	1y	2y	2y	2y
ProcedureGroup count (17)	numeric	1y	1y	2y	2y	2y
PCGroup count (45)	numeric	1y	1y	2y	2y	2y
ProviderID count (14700)	numeric	1y	1y		2y	2y
Vendor count (6388)	numeric	1y	1y		2y	2y
PCP count (1360)	numeric	1y	1y		2y	2y

PlaceSvc$\times$Specialty count (83)	numeric	1y	1y		2y	2y
PlaceSvc$\times$ProcedureGroup count (116)	numeric	1y	1y		2y	2y
PlaceSvc$\times$PCGroup count (320)	numeric	1y	1y		2y	2y
Specialty$\times$ProcedureGroup count (163)	numeric	1y	1y		2y	2y
Specialty$\times$PCGroup count (471)	numeric	1y	1y		2y	2y
ProcedureGroup$\times$PCGroup count (609)	numeric	1y	1y		2y	2y
Specialty distinctive count	numeric	1y	1y	2y	2y	2y
PlaceSvc distinctive count	numeric	1y	1y	2y	2y	2y
PCGroup distinctive count	numeric	1y	1y	2y	2y	2y
ProcedureGroup distinctive count	numeric	1y	1y	2y	2y	2y
ProviderID distinctive count	numeric	1y	1y		2y	2y
Vendor distinctive count	numeric	1y	1y		2y	2y
PCP distinctive count	numeric	1y	1y		2y	2y
PlaceSvc ratio (9)	numeric	1y	1y	2y	2y	2y
Specialty ratio (13)	numeric	1y	1y	2y	2y	2y
PCGroup ratio (46)	numeric	1y	1y	2y	2y	2y
ProcedureGroup ratio (18)	numeric	1y	1y	2y	2y	2y
DSFS min	numeric	1y	1y		2y	2y
DSFS max	numeric	1y	1y	2y	2y	2y
DSFS average	numeric	1y	1y		2y	2y
DSFS standard deviation	numeric	1y	1y		2y	2y
DSFS range	numeric	1y	1y		2y	2y
DSFS$>k$?: $k=1,2,\ldots,11$ (11)	binary flag			2y		
CharlsonIndex min	numeric	1y	1y		2y	2y
CharlsonIndex max	numeric	1y	1y	2y	2y	2y
CharlsonIndex average	numeric	1y	1y		2y	2y
CharlsonIndex.standard deviation	numeric	1y	1y		2y	2y
CharlsonIndex range	numeric	1y	1y		2y	2y
CharlsonIndex sum	numeric	1y	1y			
claim counts	numeric	1y	1y	2y	2y	2y
LengthOfStay min	numeric	1y	1y		2y	2y
LengthOfStay max	numeric	1y	1y	2y	2y	2y
LengthOfStay avg	numeric	1y	1y		2y	2y
LengthOfStay standard deviation	numeric	1y	1y		2y	2y
LengthOfStay #missing	numeric	1y	1y		2y	2y
LengthOfStay #suppressed	numeric	1y	1y	2y	2y	2y
LengthOfStay #valid	numeric	1y	1y		2y	2y
LengthOfStay sum	numeric	1y	1y	2y	2y	2y
DrugCount min	numeric	1y	1y		2y	2y
DrugCount max	numeric	1y	1y		2y	2y
DrugCount average	numeric	1y	1y		2y	2y
DrugCount range	numeric	1y	1y		2y	2y
DrugCount #entry	numeric	1y	1y	2y	2y	2y
DrugCount sum	numeric	1y	1y	2y	2y	2y
DrugCount velocity	numeric	1y	1y			
DSFS-in-DrugCount$>k$?: $k=1,\ldots,11$ (11)	binary flag			2y		
LabCount.min	numeric	1y	1y		2y	2y
LabCount.max	numeric	1y	1y		2y	2y
LabCount.avg	numeric	1y	1y		2y	2y
LabCount.range	numeric	1y	1y		2y	2y

LabCount.#entry	numeric	1y	1y	2y	2y	2y
LabCount.sum	numeric	1y	1y	2y	2y	2y
LabCount.velocity	numeric	1y	1y			
DSFS-in-LabCount$>k$?: $k=1,\ldots,11$ (11)	binary flag			2y		
AdmissionRiskL70.max	numeric	1y	1y	2y	2y	2y
AdmissionRiskL70.avg	numeric	1y	1y		2y	2y
AdmissionRiskG70.max	numeric	1y	1y	2y	2y	2y
AdmissionRiskG70.avg	numeric	1y	1y		2y	2y
AdmissionRiskL70.sum	numeric	1y	1y	2y	2y	2y
AdmissionRiskG70.sum	numeric	1y	1y	2y	2y	2y
PlaceSvc$\times$LengthOfStay (54)	numeric		1y			2y
Specialty$\times$LengthOfStay (38)	numeric		1y			2y
ProcedureGroup$\times$LengthOfStay (119)	numeric		1y			2y
PCGroup$\times$LengthOfStay (290)	numeric		1y			2y
PlaceSvc$\times$DSFS (96)	numeric		1y			2y
Specialty$\times$DSFS (144)	numeric		1y			2y
ProcedureGroup$\times$DSFS (204)	numeric		1y			2y
PCGroup$\times$DSFS (539)	numeric		1y			2y
LengthOfStay$\times$DSFS (119)	numeric		1y			2y

A.2.2 Features mainly meant for linear models: $\ell 1$ and $\ell 2$

Feature set: $\ell 1$ and $\ell 2$

Description	Discretization interval	Type	$\ell 1$	$\ell 2$
PlaceSvc count (8)		numeric	1y	
Specialty count (12)		numeric	1y	
ProcedureGroup count (17)		numeric	1y	
PCGroup count (45)	N/A	numeric	1y	
ProviderID count (14700)		numeric	1y	
Vendor count (6388)		numeric	1y	
PCP count (1360)		numeric	1y	
PlaceSvc$\times$Specialty count (83)		numeric	1y	
PlaceSvc$\times$ProcedureGroup count (116)		numeric	1y	
PlaceSvc$\times$PCGroup count (320)	N/A	numeric	1y	
Specialty$\times$ProcedureGroup count (163)		numeric	1y	
Specialty$\times$PCGroup count (471)		numeric	1y	
ProcedureGroup$\times$PCGroup count (609)		numeric	1y	
DSFS count (12)		numeric	1y	
CharlsonIndex count (4)		numeric	1y	
LengthOfStay count (11)	N/A	numeric	1y	
DSFS-in-DrugCount count (12)		numeric	1y	
DSFS-in-LabCount count (12)		numeric	1y	
Specialty distinctive count$>k$? (8)		binary flag	1y	
PlaceSvc distinctive count$>k$? (8)		binary flag	1y	
PCGroup distinctive count$>k$? (8)		binary flag	1y	
ProcedureGroup distinctive count$>k$? (8)		binary flag	1y	
ProviderID distinctive count$>k$? (8)	$k=1,2,3,4,5,10,15,20$	binary flag	1y	
Vendor distinctive count$>k$? (8)		binary flag	1y	
PCP distinctive count$>k$? (8)		binary flag	1y	

Vendor distinctive count> k? (8)		binary flag	1y	
PCP distinctive count> k? (8)		binary flag	1y	
#claim> k? (10)	$k = 1, 2, 3, 4, 5, 10, 15, 20, 30, 40$	binary flag	1y	
DSFS.max> k? (12)	$k = 1, 2, \ldots, 12$	binary flag	1y	
CharlsonIndex.max> k? (4)	$k = 0, 2, 4, 6$	binary flag	1y	
DSFS-in-DrugCount.max> k? (12)	$k = 1, 2, \ldots, 12$	binary flag	1y	
DSFS-in-LabCount.max> k? (12)		binary flag	1y	
DrugCount.#entry> k? (7)	$k = 0, 1, 2, 4, \ldots, 10$	binary flag	1y	
LabCount.#entry> k? (7)		binary flag	1y	
DrugCount.sum> k? (10)	$k = 0, 1, 3, 5, 10, 20, \ldots, 60$	binary flag	1y	
LabCount.sum> k? (12)	$k = 0, 1, 3, 5, 10, 20, \ldots, 80$	binary flag	1y	
{Male|Female|?}×PlaceSvc count (24)	N/A	numeric		1y
{Male|Female|?}×Specialty count (36)		numeric		1y
{Male|Female|?}×ProcedureGroup count (51)		numeric		1y
{Male|Female|?}×PCGroup count (135)		numeric		1y
(Age> k?)×{Male|Female|?} (27)	$k = 0, 10, \ldots, 80$	binary flag		1y
(Age> k?)×PlaceSvc count (72)		numeric		1y
(Age> k?)×Specialty count (108)		numeric		1y
(Age> k?)×ProcedureGroup count (153)		numeric		1y
(Age> k?)×PCGroup count (405)		numeric		1y

A.2.3 Feature singletons: DiH, CT, CY

The feature sets defined below consist of one feature.

Feature set	Description	type
DiH	DaysInHospital in the previous year; -1 if unknown.	numeric
CT	ClaimsTruncated in the previous year; -1 if unknown.	$\{1, 0, -1\}$
CY	Count of years in which the member has any claim	numeric

A.2.4 Feature set: p1

Feature set p1 is an extension of pcp_prob of [4]. Let $f(x)$ denote the value of feature f for the member x, and let $d(x)$ be member x's DaysInHospital in the next year. Considering the probability that $d(\cdot) > 0$ conditioned on the value of $f(\cdot)$ according to distribution D, we define

$$q_f(m) = \begin{cases} \Pr(d(x) > 0 \mid f(x) = 0)_{x \sim D} & \text{if } f(m) = 0 \\ \Pr(d(x) > 0 \mid f(x) \neq 0)_{x \sim D} & \text{otherwise} \end{cases}$$

Let F be a set of count features derived from the same original field, e.g., a set of eight count features derived from PlaceSvc. Then for each member m, we derive the following three features:

$$\max_{f \in F} q_f(m) \ , \quad \min_{f \in F} q_f(m) \ , \quad \sum_{f \in F} q_f(m) \ .$$

The conditional probabilities were estimated from the Y1 data with the Y2 target without smoothing. To cope with the issue of rare events, we merged the features whose count of the event (either $f(m) = 0$ or $f(m) \neq 0$) is less than 10 into one feature only for this purpose. We applied this procedure to PlaceSvc, Specialty, ProcedureGroup, PrimaryConditionGroup, bigrams of these four, Vendor, ProviderID, and PCP, which resulted in 39 features.

A.3 ASO features: aso

We had four instances of ASO. In instance#1 and #2, the auxiliary tasks are in the form of predicting one view (*target view*) based on another view (*feature view*) as follows:

- Instance#1. Target views: Table 3. Feature views: Table 4.
- Instance#2. Target views: Table 3. Feature views: Table 6.
  The pairs of target view and feature view that overlap (e.g., 'PlaceSvc counts' and 'PlaceSvc$\times$Specialty counts') are excluded.

In #3 and #4, the prediction targets of the auxiliary tasks are the DaysInHospital predictions made by one view (*indirect-target view*), and the features are another view (*feature view*).

- Instance#3. Indirect-target views: Table 5. Feature views: Table 4.
- Instance#4. Indirect-target views: Table 5. Feature views: views in Table 6.
  The pairs of indirect-target view and feature view that overlap are excluded.

Note that construction of auxiliary problems in instance#1 and #2 follows the *unsupervised strategy* whereas that of #3 and #4 follows the *partially-supervised strategy*; these strategies are discussed in Sections 4.2 of [1]. The auxiliary tasks in #3 and #4 require as preprocessing training linear predictors to predict DaysInHospital from the indirect target views. For the training on the auxiliary tasks (and preprocessing), all the Y1–Y3 data was used in instance#1 and #2, and Y1 data was used with Y2 DaysInHospital in #3 and #4. The L_2 regularization parameter was set to 0.001. The dimensionality for these four ASO instances were set to 70, 50, 11, and 12, respectively. We call the concatenation of the features generated by the four instances (therefore of dimensionality 143=70+50+11+12) feature set "aso".

PlaceSvc counts
Specialty counts
ProcedureGroup counts
PCGroup counts
all the features in m1
all the features derived from CharlsonIndex in t1
all the features derived from LengthOfStay in t1
all the AdmissionRisk features in t1
all the features derived from DSFS in t1
all the features derived from DrugCount in t1
all the features derived from LabCount in t1

Table 3: Target views for ASO instance#1 and #2.

Vendor counts
ProviderID counts
PCP counts

Table 4: Feature views for ASO instance#1 and #3. One view per line.

all the features derived from PlaceSvc in $\ell1$
all the features derived from Specialty in $\ell1$
all the features derived from ProcedureGroup in $\ell1$
all the features derived from PCGroup in $\ell1$
all the features in m1all the features derived from CharlsonIndex in $\ell1$
all the features derived from LengthOfStay in $\ell1$
all the features derived from DSFS in $\ell1$
all the features derived from DrugCount in $\ell1$
all the features derived from LabCount in $\ell1$
all the features in $\ell1$ excluding those derived from Vendor, ProviderID, or PCP

Table 5: Indirect-target views for ASO instance#3 and #4. One view per line.

PlaceSvc$\times$Specialty counts
PlaceSvc$\times$ProcedureGroup counts
PlaceSvc$\times$PCGroup counts
Specialty$\times$ProcedureGroup counts
Specialty$\times$PCGroup counts
ProcedureGroup$\times$PCGroup counts
PlaceSvc$\times$LengthOfStay counts
Specialty$\times$LengthOfStay counts
ProcedureGroup$\times$LengthOfStay counts
PCGroup$\times$LengthOfStay counts
PlaceSvc$\times$DSFS counts
Specialty$\times$DSFS counts
ProcedureGroup$\times$DSFS counts
PCGroup$\times$DSFS counts
LengthOfStay$\times$DSFS counts

Table 6: Feature views for ASO instance#2 and #4. One view per line.

B Datasets

Datasets, which were the actual input to training and application of the models, were generated from the feature sets introduced above. There are 21 types of datasets. Some runs used them without change, and some runs added or removed certain features, which will be described later.

Notation

- The *target year* is the year associated with DaysInHospital used as either the training target or test target (i.e., the values we predict). The *test target year* is always Y4. The *training target year* is shown in the table.
- Y_{-1} indicates the year before the target year, and Y_{-2} indicates the year two years before the target year. For example, when Y3 is the training target year, Y_{-1} is Y2 and Y_{-2} is Y1. To apply the models to the test data, the test target year is Y4, therefore, Y_{-1} is Y3 and Y_{-2} is Y2.
- For easier understanding, consider the feature sets such as t1 and t3 to be *feature generators* applied to the claim data. For example, t1(Y_{-1}) below means application of feature generator t1 to the claim data in year Y_{-1}. w1(Y_{-1},Y_{-2}) applies w1 to the claim data in the two-year period from Y_{-2} to Y_{-1}.

Dataset names	Dataset content	Training target year
t1	m1+t1(Y_{-1})	Y3
t1w1	m1+t1(Y_{-1})+w1(Y_{-1},Y_{-2})	Y3
t1w2	m1+t1(Y_{-1})+w2(Y_{-1},Y_{-2})	Y3
t1w3	m1+t1(Y_{-1})+w3(Y_{-1},Y_{-2})	Y3
t1t1	m1+t1(Y_{-1})+t1(Y_{-2})	Y3
t1t1w1	m1+t1(Y_{-1})+t1(Y_{-2})+w1(Y_{-1},Y_{-2})	Y3
t1t1w2	m1+t1(Y_{-1})+t1(Y_{-2})+w2(Y_{-1},Y_{-2})	Y3
t1t1w3	m1+t1(Y_{-1})+t1(Y_{-2})+w3(Y_{-1},Y_{-2})	Y3
t1*	m1+t1(Y2); m1+t1(Y1)	Y3;Y2
t3	m1+t3(Y_{-1})	Y3
t3w1	m1+t3(Y_{-1})+w1(Y_{-1},Y_{-2})	Y3
t3w2	m1+t3(Y_{-1})+w2(Y_{-1},Y_{-2})	Y3
t3w3	m1+t3(Y_{-1})+w3(Y_{-1},Y_{-2})	Y3
t3t3	m1+t3(Y_{-1})+t3(Y_{-2})	Y3
t3t3w1	m1+t3(Y_{-1})+t3(Y_{-2})+w1(Y_{-1},Y_{-2})	Y3
t3t3w2	m1+t3(Y_{-1})+t3(Y_{-2})+w2(Y_{-1},Y_{-2})	Y3
t3t3w3	m1+t3(Y_{-1})+t3(Y_{-2})+w3(Y_{-1},Y_{-2})	Y3
t3*	m1+t3(Y2); m1+t1(Y1)	Y3;Y2
u1	m1+ℓ1(Y_{-1})+ℓ2(Y_{-1})	Y3
u1*	m1+ℓ1(Y2)+ℓ2(Y2); m1+ℓ1(Y1)+ℓ2(Y1)	Y3;Y2
u1u1	m1+ℓ1(Y_{-1})+ℓ2(Y_{-1})+ℓ1(Y_{-2})+ℓ2(Y_{-2})	Y3

Note that the number of data points in t1*, t3*, and u1* is about twice that of the other datasets, and they serve only as training data with a concatenation of Y3 target and Y2 target. In these datasets, there are two data points for the members who have claims in both years. Other datasets serve as training data with the Y3 target and as test data for predicting the Y4 target.

C Runs

Each of the five runs shown in Section 4 is a blend of multiple runs. Tables 7–12 show the individual runs blended into those five runs.

Notation $F_{<n}$ denotes a set of features that have non-zero values for the data points fewer than n.

Datasets	Add ...	Remove ...	Method
t1, t1w1, t1w2, t1w3, t1t1, t1t1w1, t1t1w2, t1t1w3, t1*	None	None	RGF, GBDT
t3, t3w1, t3w2, t3w3, t3t3, t3t3w1, t3t3w2, t3t3w3, t3*	None	None	RGF, GBDT
t3w1, t3w3, t3t3, t3t3w1, t3t3w3	DiH & p1	$F_{<10}$	RGF, GBDT
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	DiH	AdmRisk & $F_{<100}$	RGF
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	DiH	LoS.#supp & $F_{<100}$	RGF
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	DiH	UnkGender & $F_{<100}$	RGF
u1*	None	None	RGF, GBDT
u1u1	None	$F_{<10}$	RGF, GBDT
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	DiH	$F_{<100}$	RGF, GBDT

Table 7: Run#1 part1/2: RGF and GBDT

Datasets	Remove ...	Base	Parameters for random forests
t3, t3t3, t3w1, t3w3	$F_{<10}$	R(t1*), G(t1*)	$m = 50,\ r = 0.33,\ n = 50$
t1w1, t1w2, t1t1w1, t1t1w2	None	R(t1*), R(t3t3w3)	$m = 100,\ r = 0.3,\ n = 50$
t1w3, t1t1w3	None	R(t1*), R(t3t3w3)	$m = 100,\ r = 0.2,\ n = 50$
t3w1, t3w2, t3w3, t3t3w1, t3t3w2, t3t3w3	None	R(t1*), R(t3t3w3)	$m = 100,\ r = 0.2,\ n = 50$

Table 8: Run#1 part2/2: Random forests trained for residual of RGF or GBDT. m: the minimum training data points per node; r: feature sampling ratio; n: the number of trees. No data sampling. In the 'Base' column, R(x) is an RGF run applied to dataset x, and G(x) is a GBDT run applied to dataset x.

Datasets	Add ...	Remove ...	Method
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	DiH	$F_{<100}$	RGF, GBDT
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	DiH& CT	$F_{<100}$	RGF, GBDT
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	CT	$F_{<100}$	RGF, GBDT

Table 9: Run#2: RGF and GBDT

Datasets	Add.	Rmv.	Conversion	Base
u1	aso	$F_{<10}$	$x \leftarrow \log(x+1)$	R(t1*), G(t1*), R(t3*), G(t3*), R(t3t3w3), G(t3t3w3)
u1*	aso	$F_{<10}$	$x \leftarrow \log(x+1)$	R(t1*), G(t1*), R(t3*), G(t3*)
u1	aso	$F_{<10}$	$x \leftarrow \min(1, \log(x+1))$	R(t1*), G(t1*), R(t3*), G(t3*), R(t3t3w3), G(t3t3w3)
u1*	aso	$F_{<10}$	$x \leftarrow \min(1, \log(x+1))$	R(t1*), G(t1*), R(t3*), G(t3*)
u1, u1*	–	$F_{<10}$	$x \leftarrow \min(1, \log(x+1))$	R(t1*), G(t1*), R(t3*), G(t3*)
u1, u1*	–	$F_{<10}$	$x \leftarrow \log(x+1)$	R(t1*), G(t1*), R(t3*), G(t3*)

Table 10: Run#3: Linear models trained for residual of RGF or GBDT. In the 'Base' column, R(x) is an RGF run applied to dataset x, and G(x) is a GBDT run applied to dataset x.

Datasets	Add ...	Remove ...	Method
t3w1, t3w3, t3t3, t3t3w1, t3t3w3	CY& DiH& p1	$F_{<100}$	RGF
t1w1, t1w3, t1t1, t1t1w1, t1t1w3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	CY	$F_{<10}$	RGF, GBDT

Table 11: Run#4: RGF and GBDT. For each combination of datasets and methods, two runs were performed: one with model size 10000 (in terms of the number of leaf nodes in the forest) and the other with model size 20000.

Datasets	Add ...	Remove ...	Method
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	DiH& CT	$F_{<100}$	RGF, GBDT
t3, t3w1, t3w3, t3t3, t3t3w1, t3t3w3	CT	$F_{<100}$	RGF, GBDT

Table 12: Run#5 RGF and GBDT