My milestone 1 solution to the Heritage Health Prize

Willem Mestrom

September 21th, 2011
(revised November 14th, 2011)

1 Introduction

My milestone 1 solution to the Heritage Health Prize with a RMSLE score of 0.457239 on the leaderboard consists of a linear blend of 21 result. These are mostly generated by relatively simple models which are all trained using stochastic gradient descent. First in section 2 I provide a description of the way the data is organized and the features that were used. Then in section 3 the training method and the post-processing steps are described. In section 4 each individual model is briefly described, all the relevant meta-parameter settings can be found in appendix Parameter settings. Finally the weights in the final blend are given in section 5.

2 The data

2.1 Data organization

Most models are build on only the release 2 data. In this dataset there is some basic information about the members like sex and age and there is claim data available for three years: Y1, Y2 and Y3. Finally there is Days-In-Hospital (DIH) data for Y2 and Y3. The goal is to predict Y4 DIH. One way to build a prediction model is using the following 'one-year-history' setup:

One year claims	
Training	member data, Y1 claim data → Y2 DIH
	member data, Y2 claim data → Y3 DIH
Prediction	member data, Y3 claim data → Y4 DIH

Here one year of claim data is used to predict the days in hospital for next year. This organization has as disadvantage that you make the final Y4 predictions based only on the claim data of Y3. An alternative is the following 'two-year-history' setup:

Two years claims	
Training	member data, Y1 & Y2 claim data, Y2 DIH → Y3 DIH
Prediction	member data, Y2 & Y3 claim data, Y3 DIH → Y4 DIH

Using this data organisation there is more data available for the Y4 DIH prediction but only Y3 DIH to learn from instead of Y2 and Y3. Nineteen models are build using the 'one-year-history' data organization, two models (SigCatVec3c-Y3 and SigCatVec4) are build using the 'two-year-history' data organisation. Since I don't have a lot of experience with the second setup I can't say whether one is more effective then the other but using both certainly helps in the blend.

The last model (SigClaimVec7) also use the drug and lab data provided with the release 3 data. This data is also used using the 'one-year-history' organisation.

2.2 Features

Some of the columns in the data files contain numeric values, others text values. Many columns also have some missing values. In table 1 all used columns are listed. It also lists the number of categories found for each column. Some models use each claim record separately, others use the set of distinct categories over all claim records for a member in a particular year. Some models also use the number of times a category occurs for a

File	Column	Number of categories	MC0	MC1	MC2	MC3
Members	AgeAtFirstClaim	10	●	●	●	●
	Sex	3	●	●	●	●
DaysInHospital	ClaimsTrunctated	2				
	DaysInHospital	16				● (previous year)
Claims	ProviderID	14700			●	
	Vendor	6388			●	
	PCP	1360			●	
	PCP (last claim)	1360			●	
	Specialty	13	●	●	●	●
	PlaceSvc	9	●	●	●	●
	PayDelay	- (not used)				
	LengthOfStay	11		●	●	●
	DSFS	13		●	●	●
	PrimaryConditionGroup	46		●	●	●
	CharlsonIndex	6		●	●	●
	ProcedureGroup	18		●	●	●
	SupLOS	2		●	●	●
DrugCount	DrugCount	7				
LabCount	LabCount	10				

Table 1: Data columns

member. In the models three such sets are used, $MC1_m$, $MC2_m$ and $MC3_m$ for each member m. The columns used to build each of these sets are listed in the table 1 in the columns MC1, MC2 and MC3.

$MC0$ is provided for an example: suppose we have a member x with age=40, sex=male who has two claims, claim 1 with Specialty=Emergency, PlaceSvc=Urgent Care and claim 2 with Specialty=Diagnostic Imaging, PlaceSvc=Urgent Care. This gives the following set and counts:

$$MC0_x = \{40,\ male,\ emergency,\ diagnostic\,imaging,\ urgent\,care\}$$

$$\begin{aligned} count_{x,40} &= 1 \\ count_{x,male} &= 1 \\ count_{x,emergency} &= 1 \\ count_{x,diagnostic\,imaging} &= 1 \\ count_{x,urgent\,care} &= 2 \end{aligned}$$

In the describtions of the models the following variables are used to reference the columns in the data files:

sex_m	sex of member m
age_m	age of member m
$truncated_m$	1 if claims for member m were truncated, 0 otherwise
$nclaims_m$	number of claims for member m
$provider_c$	provider for claim c
$vendor_c$	vendor for claim c
pcp_c	primary care physician for claim c
$specialty_c$	specialty for claim c
$place_c$	place of service for claim c
los_c	length of stay for claim c
$dsfs_c$	days since first service for claim c
$dtls_c$	'days till last service' (maximum days since first service for the member minus days since first service) for claim c
pcg_c	primary condition group for claim c
$charlsonIndex_c$	Charlson index for claim c
pg_c	procedure group for claim c
$suplos_c$	1 if length of stay for claim c was truncated, 0 otherwise

3 Training and predicting

3.1 Training method

All models are trained to model $\ln(DIH + 1)$ instead of DIH. This simplifies the RMSLE scoring measure to the more standard RMSE. Only when a submission file is generated the $\ln(DIH + 1)$ values are converted back to DIH values.

The models are trained using a stochastic gradient descent[1] without mini-batches. For each parameter to learn there is a learning rate η and a shrinkage parameter λ. For each training case (a member, year combination) all applicable parameters are updated using the update rule: $f_i \leftarrow (1 - \eta\lambda) \cdot f_i + \eta \cdot gradient$.

The number of iterations through the dataset is not the same for all models. A common approach is to stop as soon as the score for a validation set starts to increase. I have taken a different approach; the learning rates are optimized for a fixed number of iterations. The iterations are split up in a number of phases, each phase has its own set of learning rates. For some models this leads to a large number of learning rates to optimize. This optimization process is vital to get good results but doing this manually is extremely time consuming. Therefore I used some automated procedures for the optimization process (along with some manual tuning). The methods used were Nelder-Mead[2] and a simplified Rosenbrock algorithm[3]. The rotation of the coordinate system that is used in this algorithm turned out to be ineffective in most cases so this function was removed from the algorithm. Also the the step sizes were adjusted to fit this particular problem. When a change is succesful the stepsize is multiplied by 1.3, when a change is not succesful the stepsize is multiplied by -0.5 and the initial stepsize is 0.1 times the current parameter value. When convergence of the automated procedure was going very slowly even though the accuracy was not near the expected optimum (close to the accuracy of a similar model) I set one or a few of the model parameters to a very different value to get out of the local minimum or plateau. After such a manual intervention the automated procedure was continued. The final model parameters for each model can be found in appendix A.

For each model in the section 4 all the learning rates are given for each phase. If the λ parameter is omitted the value is 0. The parameter values are written in the full numerical precision as they were used. Most of the time only the first two or three digits are significant, the rest is only included for completeness.

3.2 Parameter initialization

The models use both scalar parameters and vector parameters. Scalar parameters are initialized to zero unless specified otherwise. For vectors it is a bit different. Vectors are usually used in a product with an other vector

[1] http://en.wikipedia.org/wiki/Stochastic_gradient_descent
[2] http://en.wikipedia.org/wiki/Nelder-Mead_method
[3] http://www.applied-mathematics.net/optimization/rosenbrock.html

parameter. When all vectors are initialized to zero all gradients would be zero and the gradient descent would be stuck right at the initial state. So at least one of the vectors in a vector product should be none zero. Therefore all elements of each vectors are initialized using samples from a uniform random distribution between -0.01 and +0.01 unless stated otherwise (some experimentation showed that for some vectors an initial value of $\vec{0}$ or $\vec{1}$ gives better results).

3.3 Making predictions

Each model can be used directly to make predictions for Y4 DIH. In practice however many models generate predictions with too much variance. To get better predictions the variance can be reduced by averaging the predictions of the model when trained on several different subsets of the data. Each of these subsets is used to generate a complete Y4 DIH prediction. The left-out set is always non-overlapping, so each data point is only excluded once. The final prediction for the model is the arithmetic mean of the $\ln(DIH + 1)$ predictions for each of the training runs.

For each model exactly one of the methods in table 2 is used. The first three use only a single training run and may produce predictions with somewhat high variance. The last two method use multiple runs whose results are then averaged, these methods produce predictions with lower variance.

Method	**Runs**	**Data used per run**
Qualifying	1	100%
Qualifying (Y1 only)	1	50% (using only Y1)
Qualifying (70%)	1	70% (30% never used)
Qualifying (CV 4)	4	75%
Qualifying (CV 10)	10	90%

Table 2: Prediction methods

3.4 Post processing

For some models an additional post-processing step is used. When a model predicts an extreme value it is almost always a good idea to adjust this prediction towards the mean. This idea was effectively used by Edward de Grijs in the Netflix Prize competition and proved to be useful here as well. The formula used is:

$$\begin{aligned} \tilde{p_m} &= min(cc_{max}, max(cc_{min}, cc_{bias} + cc_{slope} \cdot p_m)) \\ \hat{p_m} &= 0.5\,(2\tilde{p_m})^{(cc_a + cc_b \cdot \tilde{p_m} + cc_c \cdot \tilde{p_m} \cdot \tilde{p_m})} \end{aligned}$$

Where p_m is the original prediction of the model and $\hat{p_m}$ is the final prediction. cc_{bias}, cc_{slope}, cc_{min}, cc_{max}, cc_a, cc_b and cc_c are parameters which are optimized using the simplified Rosenbrock algorithm. The used parameter values can be found in appendix A.

4 The models

4.1 CatVec1

The CatVec1 model learns two feature vectors of dimension 4 per distinct category in the $MC2$ set.

f_i vector of dimension 4 for category i
g_i vector of dimension 4 for category i

$$p_m = \left(\sum_{i \in MC2_m} f_i\right)^T \left(\sum_{i \in MC2_m} g_i\right)$$

The summation is over the elements in the set $MC2_m$, so if member m has an age 50-59 then the set will include "AgeAtFirstClaim=50-59" and the summation will include this category. If the member has a different age the set will not include "AgeAtFirstClaim=50-59" and the summation will not include this category.

To illustrate the stochastic gradient descent the complete update rules for this model are given here:

$$
\begin{aligned}
e_m &= \ln(1 + DIH_m) - p_m \\
\hat{f}_i &= (1 - \lambda_f \eta_f) f_i + \eta_f e \left(\sum_{j \in MC2_m} g_j \right) \\
\hat{g}_i &= (1 - \lambda_g \eta_g) g + \eta_g e \left(\sum_{j \in MC2_m} f_j \right) \\
f_i &\leftarrow \hat{f}_i \\
g_i &\leftarrow \hat{g}_i
\end{aligned}
$$

4.2 CatVec2

This model is identical to the CatVec1 model except for the parameter settings.

4.3 CatVec3

This model is similar to the CatVec1 model but this time a log function is added.

$$
p_m = \ln \left(1 + \left(\sum_{i \in MC2_m} f_i \right)^T \left(\sum_{i \in MC2_m} g_i \right) \right)
$$

Note the for this model the log function is not used in the calculation of the gradients, i.e. the update rules are identical to the update rules for CatVec1.

4.4 SigCatVec1

The model uses one feature vector of dimension 12 per distinct category. First these applicable vectors are summed. After that a sigmoid transformation is applied to each of the elements of the sum vector. Finally a single 'score' vector s is used as a weighting for each of the vector elements.

$$
p_m = s^T \sigma \left(\sum_{i \in MC2_m} f_i \right)
$$

Where σ is the sigmoid function defined as $\sigma(x) = \frac{1}{1+e^{-x}}$.

4.5 SigCatVec2

This model is identical to the SigCatVec1 model except that the vector dimension is set to 40 for this model.

4.6 SigCatVec3a

This model is similar to theSigCatVec1 model but adds a factor for the number of occurences of each category within the member.

$$
p_m = s^T \sigma \left(\sum_{i \in MC1_m} (f_i + g_i \cdot count_{m,i}) \right)
$$

4.7 SigCatVec3b

This model is the same as the SigCatVec3a model but uses different parameter settings.

4.8 SigCatVec3c-Y3

This model is similar to the SigCatVec3a model but for this model the 'two-year-history' data organisation is used as described in section 2.1. Due to a bug in my code this model generated a prediction for Y3 which was used as the prediction for Y4.

4.9 SigCatVec4

This model is similar to the SigCatVec3c-Y3 model but without the bug and using the $MC3$ set instead of the $MC1$ set.

$$p_m \quad = \quad s^T \sigma \left(\sum_{i \in MC3_m} (f_i + g_i \cdot count_{m,i}) \right)$$

4.10 SigCatVec5

This model is similar to theSigCatVec3a model but adds an additional weighting vector h. This vector is initialized by setting each element to 1.

$$p_m \quad = \quad s^T \sigma \left(\sum_{i \in MC1_m} (f_i + g_i \cdot count_{m,i}) \circ h_i \right)$$

Where $\circ$ denotes the Hadamard (pointwise) product[4].

4.11 SigCatVec6

This model is similar to theSigCatVec5 model except that the square root of the count is used.

$$p_m \quad = \quad s^T \sigma \left(\sum_{i \in MC1_m} (f_i + g_i \sqrt{count_{m,i}}) \circ h_i \right)$$

4.12 SigCatVec7

This model is similar to theSigCatVec6 model except that set MC2 is used instead of set MC1.

$$p_m \quad = \quad s^T \sigma \left(\sum_{i \in MC2_m} (f_i + g_i \sqrt{count_{m,i}}) \circ h_i \right)$$

4.13 SigCatVec8

This model is the same as the SigCatVec7 model but with different parameters and an additional postprocessing step as described in section Post processing.

[4] http://en.wikipedia.org/wiki/Matrix_multiplication

4.14 PerClaim

This model is very different from the previous models. In the previous models all variables (sex, age, place, specialty, etc) were traited equaly. In this model some variables have different learning rates and weightings then others. For example the claim parameters are scaled with the inverse square root of the number of claims but the member parameters are not scaled. In general variables are treated differently based on their meaning. The table below lists all the learned parameters. (Note: the double claim bias was an error and should not have a positive effect on the result).

Variable	Description	Learning rate
ms_i	scalar for member sex i	η_1
ma_i	scalar for member age i	η_1
mt	scalar	η_1
$cbias1$	scalar	η_1
$cbias2$	scalar	η_1
cpr_i	scalar for claim provider i	η_1
cv_i	scalar for claim vendor i	η_1
$cpcp_i$	scalar for claim primary care physician i	η_1
cs_i	scalar for claim specialty i	η_1
cpl_i	scalar for claim place i	η_1
cl_i	scalar for claim los i	η_1
$cpcg_i$	scalar for claim primary condition group i	η_1
$charlson$	scalar	η_1
cpg_i	scalar for claim procedure group i	η_1
csl_i	scalar for claim suplos i	η_1
$isp_{i,j}$	scalar for claim combination of specialty i and primary condition group j	η_1
$ia_{i,j}$	scalar for claim combination of age i and primary condition group j	η_1
$is_{i,j}$	scalar for claim combination of sex i and primary condition group j	η_1
$ipl_{i,j}$	scalar for claim combination of place i and primary condition group j	η_1
$vms1_i$	vector for member sex i	η_2
$vms2_i$	vector for member sex i	η_3
$vma1_i$	vector for member age i	η_2
$vma2_i$	vector for member age i	η_3
$vcs1_i$	vector for claim specialty i	η_2
$vcs2_i$	vector for claim specialty i	η_3
$vcp1_i$	vector for claim place of service i	η_2
$vcp2_i$	vector for claim place of service i	η_3
$vcpcg1_i$	vector for claim primary condition group i	η_2
$vcpcg2_i$	vector for claim primary condition group i	η_3

These parameters are combined into the following model

$$
\begin{aligned}
member_m &= ms_{sex_m} + ma_{age_m} + mt \cdot truncated_m \\
base_c &= cbias1 + cbias2 + cpr_{provider_c} + cv_{vendor_c} + cpcp_{pcp_c} + cs_{specialty_c} + cp_{place_c} + cl_{los_c} + cpcg_{pcg_c} \\
&\quad + charlson \cdot charlsonIndex_c + cpg_{pg_c} + csl_{suplos_c} \\
interaction_c &= isp_{specialty_c, pcg_c} + ia_{specialty_c, pcg_c} + is_{specialty_c, pcg_c} + ipl_{specialty_c, pcg_c} \\
claims_m &= \sum_{c \in claims_m} \frac{(base_c + interaction_c)}{\sqrt{nclaims_m}} \\
vec_m &= \left(vms1_{sex_m} + vma1_{sex_m} + \sum_{c \in claims_m} (vcs1_{specialty_c} + vcp1_{place_c} + vcpcg1_{pcg_c}) \right)^T \\
&\quad \left(vms2_{sex_m} + vma2_{sex_m} + \sum_{c \in claims_m} (vcs2_{specialty_c} + vcp2_{place_c} + vcpcg2_{pcg_c}) \right) \\
p_m &= \ln\left(1 + \operatorname{clamp}\left(mean + member_m + claims_m + vec_m\right)\right)
\end{aligned}
$$

Where the clamp function is defined as $\operatorname{clamp}(x) = \min(15,\ \max(0,\ x))$.

4.15 SigClaimVec1

This model is similar to the PerClaim model but uses vectors instead of scalars for most variables. The dimension of all vectors in this model is 12. (Note: the three claim bias vectors were accidentaly introduced, one should be sufficient). Also it adds a set of parameters to further tune the learning rates, these new parameters are fixed over the phases in order to limit the total number of parameters.

Variable	Description	Learning rate
$mbias$	vector	η_1
ms_i	vector for member sex i	η_1
ma_i	vector for member age i	η_1
$cbias1$	vector	$\eta_2 \cdot w_{bias1}$
$cbias2$	vector	$\eta_2 \cdot w_{bias2}$
$cbias3$	vector	$\eta_2 \cdot w_{bias3}$
cpr_i	vector for claim provider i	$\eta_2 \cdot w_{provider}$
cv_i	vector for claim vendor i	$\eta_2 \cdot w_{vendor}$
$cpcp_i$	vector for claim primary care physician i	$\eta_2 \cdot w_{pcp}$
cs_i	vector for claim specialty i	$\eta_2 \cdot w_{specialty}$
cpl_i	vector for claim place i	$\eta_2 \cdot w_{place}$
cl_i	vector for claim los i	$\eta_2 \cdot w_{los}$
$cdsfs_i$	vector for claim days since first service i	$\eta_2 \cdot w_{dsfs}$
$cpcg_i$	vector for claim primary condition group i	$\eta_2 \cdot w_{pcg}$
cpg_i	vector for claim proceduregroup i	$\eta_2 \cdot w_{pg}$
$isp_{i,j}$	vector for claim combination of specialty i and primary condition group j	η_3
$ia_{i,j}$	vector for claim combination of age i and primary condition group j	η_3
$is_{i,j}$	vector for claim combination of sex i and primary condition group j	η_3
$ipl_{i,j}$	vector for claim combination of place i and primary condition group j	η_3
s	vector	η_4

These parameters are combined into the following model

$$
\begin{aligned}
member_m &= mbias + ms_{sex_m} + ma_{age_m} \\
base_c &= cbias1 + cbias2 + cbias3 + cpr_{provider_c} + cv_{vendor_c} + cpcp_{pcp_c} \\
&\quad + cs_{specialty_c} + cp_{place_c} + cl_{los_c} + cpcg_{pcg_c} + cpg_{pg_c} \\
interaction_c &= isp_{specialty_c, pcg_c} + ia_{specialty_c, pcg_c} + is_{specialty_c, pcg_c} + ipl_{specialty_c, pcg_c} \\
claims_m &= \sum_{c \in claims_m} \sigma\left(base_c + interaction_c\right) \\
p_m &= \ln\left(1 + \text{clamp}\left(s^T \sigma\left(member_m + claims_m\right)\right)\right)
\end{aligned}
$$

4.16 SigClaimVec2

This model is the same as the SigClaimVec1 model except for the learning rates and an additional learning phase.

Variable	Description	Learning rate
$mbias$	vector	$\eta_1 \cdot w_{bias}$
ms_i	vector for member sex i	$\eta_1 \cdot w_{sex}$
ma_i	vector for member age i	$\eta_1 \cdot w_{age}$
$cbias1$	vector	$\eta_2 \cdot w_{bias1}$
$cbias2$	vector	$\eta_2 \cdot w_{bias2}$
$cbias3$	vector	$\eta_2 \cdot w_{bias3}$
cpr_i	vector for claim provider i	$\eta_2 \cdot w_{provider}$
cv_i	vector for claim vendor i	$\eta_2 \cdot w_{vendor}$
$cpcp_i$	vector for claim primary care physician i	$\eta_2 \cdot w_{pcp}$
cs_i	vector for claim specialty i	$\eta_2 \cdot w_{specialty}$
cpl_i	vector for claim place i	$\eta_2 \cdot w_{place}$
cl_i	vector for claim los i	$\eta_2 \cdot w_{los}$
$cdsfs_i$	vector for claim days since first service i	$\eta_2 \cdot w_{dsfs}$
$cpcg_i$	vector for claim primary condition group i	$\eta_2 \cdot w_{pcg}$
cpg_i	vector for claim proceduregroup i	$\eta_2 \cdot w_{pg}$
$isp_{i,j}$	vector for claim combination of specialty i and primary condition group j	$\eta_3 \cdot w_{isp}$
$ia_{i,j}$	vector for claim combination of age i and primary condition group j	$\eta_3 \cdot w_{ia}$
$is_{i,j}$	vector for claim combination of sex i and primary condition group j	$\eta_3 \cdot w_{is}$
$ipl_{i,j}$	vector for claim combination of place i and primary condition group j	$\eta_3 \cdot w_{ipl}$
s	vector	η_4

4.17 SigClaimVec3

This model is identical to the SigClaimVec2 model except this one has 4 phases again and adds a post-processing step as described in section 3.4.

4.18 SigClaimVec4

This model is similar to the SigClaimVec3 model but adds a few interaction variables and adds a weighting of the claims depending on the time until the last claim in the current year (dtls as described in section Features).

Variable	Description	Learning rate
$mbias$	vector	$\eta_1 \cdot w_{bias}$
ms_i	vector for member sex i	$\eta_1 \cdot w_{sex}$
ma_i	vector for member age i	$\eta_1 \cdot w_{age}$
$msa_{i,j}$	vector for member sex i and age j (initial value: $\vec{0}$)	$\eta_1 \cdot w_{sexage}$
$cbias1$	vector	$\eta_2 \cdot w_{bias1}$
$cbias2$	vector	$\eta_2 \cdot w_{bias2}$
$cbias3$	vector	$\eta_2 \cdot w_{bias3}$
cpr_i	vector for claim provider i	$\eta_2 \cdot w_{provider}$
cv_i	vector for claim vendor i	$\eta_2 \cdot w_{vendor}$
$cpcp_i$	vector for claim primary care physician i	$\eta_2 \cdot w_{pcp}$
cs_i	vector for claim specialty i	$\eta_2 \cdot w_{specialty}$
cpl_i	vector for claim place i	$\eta_2 \cdot w_{place}$
cl_i	vector for claim los i	$\eta_2 \cdot w_{los}$
$cdsfs_i$	vector for claim days since first service i	$\eta_2 \cdot w_{dsfs}$
$cpcg_i$	vector for claim primary condition group i	$\eta_2 \cdot w_{pcg}$
cpg_i	vector for claim proceduregroup i	$\eta_2 \cdot w_{pg}$
$isp_{i,j}$	vector for claim combination of specialty i and primary condition group j	$\eta_3 \cdot w_{isp}$
$ia_{i,j}$	vector for claim combination of age i and primary condition group j	$\eta_3 \cdot w_{ia}$
$is_{i,j}$	vector for claim combination of sex i and primary condition group j	$\eta_3 \cdot w_{is}$
$ipl_{i,j}$	vector for claim combination of place i and primary condition group j	$\eta_3 \cdot w_{ipl}$
$ispl_{i,j}$	vector for claim combination of sex i and place j (initial value: $\vec{0}$)	$\eta_3 \cdot w_{ispl}$
$iapl_{i,j}$	vector for claim combination of age i and place j (initial value: $\vec{0}$)	$\eta_3 \cdot w_{iapl}$
s	vector	η_4
$cdtls_i$	scalar for claim days till last service i (initial value: $\vec{1}$)	η_5

These parameters are combined into the following model

$$
\begin{aligned}
member_m &= mbias + ms_{sex_m} + ma_{age_m} \\
base_c &= cbias1 + cbias2 + cbias3 + cpr_{provider_c} + cv_{vendor_c} + cpcp_{pcp_c} \\
&\quad + cs_{specialty_c} + cp_{place_c} + cl_{los_c} + cpcg_{pcg_c} + cpg_{pg_c} \\
interaction_c &= isp_{specialty_c,pcg_c} + ia_{specialty_c,pcg_c} + is_{specialty_c,pcg_c} + ipl_{specialty_c,pcg_c} + ispl_{sex_c,place_c} + iapl_{age_c,place_c} \\
claims_m &= \sum_{c \in claims_m} cdtls_c \cdot \sigma\left(base_c + interaction_c\right) \\
p_m &= \ln\left(1 + \text{clamp}\left(s^T \sigma\left(member_m + claims_m\right)\right)\right)
\end{aligned}
$$

4.19 SigClaimVec5

This model is identical to the SigClaimVec4 model except for the parameters.

4.20 SigClaimVec6

This model is very similar to the SigClaimVec4 model but adds the claims truncated variable.

Variable	Description	Learning rate
$mbias$	vector	$\eta_1 \cdot w_{bias}$
ms_i	vector for member sex i	$\eta_1 \cdot w_{sex}$
ma_i	vector for member age i	$\eta_1 \cdot w_{age}$
$msa_{i,j}$	vector for member sex i and age j (initial value: $\vec{0}$)	$\eta_1 \cdot w_{sexage}$
mt_i	vector for member claims truncated i (initial value: $\vec{0}$)	$\eta_1 \cdot w_{truncated}$
$cbias1$	vector	$\eta_2 \cdot w_{bias1}$
$cbias2$	vector	$\eta_2 \cdot w_{bias2}$
$cbias3$	vector	$\eta_2 \cdot w_{bias3}$
cpr_i	vector for claim provider i	$\eta_2 \cdot w_{provider}$
cv_i	vector for claim vendor i	$\eta_2 \cdot w_{vendor}$
$cpcp_i$	vector for claim primary care physician i	$\eta_2 \cdot w_{pcp}$
cs_i	vector for claim specialty i	$\eta_2 \cdot w_{specialty}$
cpl_i	vector for claim place i	$\eta_2 \cdot w_{place}$
cl_i	vector for claim los i	$\eta_2 \cdot w_{los}$
$cdsfs_i$	vector for claim days since first service i	$\eta_2 \cdot w_{dsfs}$
$cpcg_i$	vector for claim primary condition group i	$\eta_2 \cdot w_{pcg}$
cpg_i	vector for claim proceduregroup i	$\eta_2 \cdot w_{pg}$
$isp_{i,j}$	vector for claim combination of specialty i and primary condition group j	$\eta_3 \cdot w_{isp}$
$ia_{i,j}$	vector for claim combination of age i and primary condition group j	$\eta_3 \cdot w_{ia}$
$is_{i,j}$	vector for claim combination of sex i and primary condition group j	$\eta_3 \cdot w_{is}$
$ipl_{i,j}$	vector for claim combination of place i and primary condition group j	$\eta_3 \cdot w_{ipl}$
$ispl_{i,j}$	vector for claim combination of sex i and place j (initial value: $\vec{0}$)	$\eta_3 \cdot w_{ispl}$
$iapl_{i,j}$	vector for claim combination of age i and place j (initial value: $\vec{0}$)	$\eta_3 \cdot w_{iapl}$
s	vector	η_4
$cdtls_i$	scalar for claim days till last service i (initial value: $\vec{1}$)	η_5
$mt2_i$	scalar for member claims truncated i	η_6

These parameters are combined into the following model

$$
\begin{aligned}
member_m &= mbias + ms_{sex_m} + ma_{age_m} + mt_{truncated_m} \\
base_c &= cbias1 + cbias2 + cbias3 + cpr_{provider_c} + cv_{vendor_c} + cpcp_{pcp_c} \\
&\quad + cs_{specialty_c} + cp_{place_c} + cl_{los_c} + cpcg_{pcg_c} + cpg_{pg_c} \\
interaction_c &= isp_{specialty_c,pcg_c} + ia_{specialty_c,pcg_c} + is_{specialty_c,pcg_c} + ipl_{specialty_c,pcg_c} + ispl_{sex_c,place_c} + iapl_{age_c,place_c} \\
claims_m &= \sum_{c \in claims_m} cdtls_c \cdot \sigma\left(base_c + interaction_c\right) \\
p_m &= \ln\left(1 + \text{clamp}\left(s^T \sigma\left(member_m + claims_m\right)\right)\right)
\end{aligned}
$$

4.21 SigClaimVec7

This model is the similar to the SigClaimVec6 model but adds the supressed length of stay, charlson index, lab count and drug count variables.

Variable	Description	Learning rate
$mbias$	vector	$\eta_1 \cdot w_{bias}$
ms_i	vector for member sex i	$\eta_1 \cdot w_{sex}$
ma_i	vector for member age i	$\eta_1 \cdot w_{age}$
$msa_{i,j}$	vector for member sex i and age j (initial value: $\vec{0}$)	$\eta_1 \cdot w_{sexage}$
mt_i	vector for member claims truncated i (initial value: $\vec{0}$)	$\eta_1 \cdot w_{truncated}$
$cbias1$	vector	$\eta_2 \cdot w_{bias1}$
$cbias2$	vector	$\eta_2 \cdot w_{bias2}$
$cbias3$	vector	$\eta_2 \cdot w_{bias3}$
cpr_i	vector for claim provider i	$\eta_2 \cdot w_{provider}$
cv_i	vector for claim vendor i	$\eta_2 \cdot w_{vendor}$
$cpcp_i$	vector for claim primary care physician i	$\eta_2 \cdot w_{pcp}$
cs_i	vector for claim specialty i	$\eta_2 \cdot w_{specialty}$
cpl_i	vector for claim place i	$\eta_2 \cdot w_{place}$
cl_i	vector for claim los i	$\eta_2 \cdot w_{los}$
$cdsfs_i$	vector for claim days since first service i	$\eta_2 \cdot w_{dsfs}$
$cpcg_i$	vector for claim primary condition group i	$\eta_2 \cdot w_{pcg}$
cpg_i	vector for claim proceduregroup i	$\eta_2 \cdot w_{pg}$
$csup_i$	vector for claim supressed length of stay i (initial value: $\vec{0}$)	$\eta_2 \cdot w_{suplos}$
cch_i	vector for claim charlson index i (initial value: $\vec{0}$)	$\eta_2 \cdot w_{charlson}$
$isp_{i,j}$	vector for claim combination of specialty i and primary condition group j	$\eta_3 \cdot w_{isp}$
$ia_{i,j}$	vector for claim combination of age i and primary condition group j	$\eta_3 \cdot w_{ia}$
$is_{i,j}$	vector for claim combination of sex i and primary condition group j	$\eta_3 \cdot w_{is}$
$ipl_{i,j}$	vector for claim combination of place i and primary condition group j	$\eta_3 \cdot w_{ipl}$
$ispl_{i,j}$	vector for claim combination of sex i and place j (initial value: $\vec{0}$)	$\eta_3 \cdot w_{ispl}$
$iapl_{i,j}$	vector for claim combination of age i and place j (initial value: $\vec{0}$)	$\eta_3 \cdot w_{iapl}$
s	vector	η_4
$cdtls_i$	scalar for claim days till last service i (initial value: $\vec{1}$)	η_5
$mt2_i$	scalar for member claims truncated i	η_6
$lcnt_i$	vector for lab count i (initial value: $\vec{0}$)	$\eta_7 \cdot w_{lab}$
$dcnt_i$	vector for drug count i (initial value: $\vec{0}$)	$\eta_7 \cdot w_{drug}$

These parameters are combined into the following model

$$
\begin{aligned}
member_m &= mbias + ms_{sex_m} + ma_{age_m} + mt_{truncated_m} \\
base_c &= cbias1 + cbias2 + cbias3 + cpr_{provider_c} + cv_{vendor_c} + cpcp_{pcp_c} \\
&\quad + cs_{specialty_c} + cp_{place_c} + cl_{los_c} + cpcg_{pcg_c} + cpg_{pg_c} + csup_{suplos_c} + cch_{charlsonIndex_c} \\
interaction_c &= isp_{specialty_c,pcg_c} + ia_{specialty_c,pcg_c} + is_{specialty_c,pcg_c} + ipl_{specialty_c,pcg_c} + ispl_{sex_c,place_c} + iapl_{age_c,place_c} \\
claims_m &= \sum_{c \in claims_m} cdtls_c \cdot \sigma\left(base_c + interaction_c\right) \\
labs_m &= \sum_{l \in labs_m} lcnt_l \\
drugs_m &= \sum_{d \in drugs_m} dcnt_d \\
p_m &= \ln\left(1 + \text{clamp}\left(s^T \sigma\left(member_m + claims_m + labs_m + drugs_m\right)\right)\right)
\end{aligned}
$$

5 Final blend

From the beginning of this contest I choose not to build a single very very good model but instead create different models each modeling the variation differently. Initially I did not expect to be required to reproduce all results almost perfectly. Therefore many of the early results could not be used. The models in the final blend are a selection of the models I could reproduce exactly. The final result is a linear combination of the log+1 predictions of all the 21 models described in section4. Unfortunately no probeset is provided in this competition. Because of the different trainingsets and prediction methods used by the different models it is hard to construct a dataset that can be used effectively for blending without introducing a bias. Therefore I choose to use the approach suggested by R. Bell and Y. Koren, and C. Volinsky in "The BellKor solution to the Netflix Prize", http://www.netflixprize.com/assets/ProgressPrize2007_KorBell.pdf, 2007. The technique comes down to performing a ridge regression[5] based on the leaderboard scores. The regularization parameter α was chosen as 0.0015 * 70492. (For a more complete describtion of the technique see section 7 of this paper: http://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf). The final weights are:

Model	**RMSLE (Leaderboard)**	**Weight**
All mean	0.486459	-0.120177096860407
CatVec1	0.475757	0.0644039235679331
CatVec2	0.466581	-0.11197527538219
CatVec3	0.466570	-0.104862676479977
SigCatVec1	0.464373	0.162280493887463
SigCatVec2	0.465728	-0.0894110617494495
PerClaim	0.464028	0.0811531937177599
SigCatVec3a	0.463635	0.0813467144179884
SigCatVec5	0.462524	0.152820826103983
SigCatVec3c-Y3	0.475019	0.229896200534371
SigCatVec4	0.464062	0.153655337312371
SigCatVec3b	0.465550	-0.124937912869077
SigCatVec7	0.464516	0.132332995001435
SigCatVec6	0.463269	-0.0801709488833528
SigClaimVec1	0.461875	-0.108108296818349
SigClaimVec2	0.461792	-0.0880275674421306
SigClaimVec3	0.460468	0.150244352469803
SigCatVec8	0.463125	0.0900888972980376
SigClaimVec4	0.461351	0.0730948061470501
SigClaimVec5	0.460345	0.131935900519871
SigClaimVec6	0.460402	0.122608906537375
SigClaimVec7	0.460564	0.200886991699764

[5] http://en.wikipedia.org/wiki/Tikhonov_regularization

A Parameter settings

A.1 CatVec1

	Phase 1	Phase 2	Phase 3
number of iterations	20	50	20
η_f	0.0002649423	0.0002470584	4.470347E-06
λ_f	0.1	0.1	0.1
η_g	0.001985118	0.003892777	1.490114E-06
λ_g	0.1	0.1	0.1
Prediction method	Qualifying		

A.2 CatVec2

	Phase 1	Phase 2	Phase 3
number of iterations	20	50	20
η_f	1.112269E-06	3.853279E-05	2.598553E-07
η_g	0.0004327906	4.473469E-06	4.490988E-08
Prediction method	Qualifying		

A.3 CatVec3

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	4.177898E-05	0.001355331	0	2.609307E-06
η_g	0.0004036372	0.0003165317	3.915297E-05	7.51259E-07
Prediction method	Qualifying			

A.4 SigCatVec1

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.001578477	0.004677286	0.02737434	0
η_s	0.0004944247	0.0001497822	0	0.0004102637
Prediction method	Qualifying			

A.5 SigCatVec2

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.316035	0.008788005	0.02413593	0.0004140852
η_s	0.0008625013	0	0	0
Prediction method	Qualifying			

A.6 SigCatVec3a

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.02075013	0.08779071	0.01074343	0.0001728464
η_g	2.772494E-05	0.0001966873	0.0002280947	1.146167E-06
η_s	0.002876806	0.02247584	0	9.332498E-06
Prediction method	Qualifying			

A.7 SigCatVec3b

	Phase 1
number of iterations	200
η_f	0.0003261481
η_g	8.437122E-05
η_s	0.0001342874
Prediction method	Qualifying

A.8 SigCatVec3c-Y3

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.0003823006	0.002848325	0.004719815	2.123389E-05
η_g	6.396919E-05	0.000264585	0.0001864179	1.758224E-06
η_s	0.0001432812	0.0005024222	0.0008601856	7.672917E-06
Prediction method	Qualifying			

A.9 SigCatVec4

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.002148289	0.08317175	0.1350184	0.001309502
η_g	0.0001728866	0.0002815352	3.191739E-05	1.296711E-08
η_s	0.001004071	0.01075159	6.94342E-08	1.916467E-10
Prediction method	Qualifying (CV 4)			

A.10 SigCatVec5

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.02445947	0.06469206	0.002567296	0.0001044714
η_g	4.443048E-06	0.000822987	0.0005028971	8.60439E-06
η_h	1.315885E-05	0.005095374	0.01023562	2.814549E-06
η_s	0.002808467	0.01118349	0.001010632	1.378307E-08
Prediction method	Qualifying			

A.11 SigCatVec6

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.02231786	0.02986512	2.271697E-06	2.929901E-06
η_g	5.368412E-05	0.008716031	2.843667E-11	4.453966E-05
η_h	4.310675E-06	6.237718E-06	0.05168699	1.182587E-05
η_s	0.0024299	0.009932901	4.058366E-07	6.896256E-07
Prediction method	Qualifying (CV 4)			

A.12 SigCatVec7

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.01983373	0.08431601	8.572736E-07	1.105661E-06
η_g	5.101598E-05	0.007684739	3.041355E-11	3.403341E-05
η_h	4.794967E-06	5.112884E-06	0.02255507	5.160553E-06
η_s	0.002150354	0.008940183	1.770982E-07	3.009375E-07
Prediction method	Qualifying (CV 4)			

A.13 SigCatVec8

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_f	0.01983373	0.08431601	8.572736E-07	1.105661E-06
η_g	5.101598E-05	0.007684739	3.041355E-11	3.403341E-05
η_h	4.794967E-06	5.112884E-06	0.02255507	5.160553E-06
η_s	0.002150354	0.008940183	1.770982E-07	3.009375E-07
cc_{bias}	0.007125622			
cc_{slope}	0.9566299			
cc_{min}	0.02596347			
cc_{max}	1.408656			
cc_a	0.9043918			
cc_b	7.510946E-08			
cc_c	0.04633662			
Prediction method	Qualifying (CV 4)			

A.14 PerClaim

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
mean	0.02948345			
η_1	8.277983E-05	0.0001205155	0.000140215	9.492154E-06
η_2	9.338494E-06	7.199409E-05	8.916834E-06	5.106591E-08
η_3	1.333333E-05	0	2.724926E-05	8.670087E-06
Prediction method	Qualifying (Y1 only)			

A.15 SigClaimVec1

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_1	0.0001052431	0.0006378752	3.563327E-05	1.354099E-05
η_2	0.02928185	0.02522659	0.02143314	0.01821492
η_3	0.0312782	0.02925104	0.02465841	0.007686132
η_4	0.002197431	0.003806778	0.02130464	3.2912E-05
w_{bias1}	0.8210818			
w_{bias2}	0.9595			
w_{bias3}	0.935			
$w_{provider}$	0.6516637			
w_{vendor}	1.652997			
w_{pcp}	0.1457351			
$w_{specialty}$	1.2495			
w_{place}	2.395576			
w_{los}	0.5993496			
w_{dsfs}	1.202			
w_{pcg}	1.652592			
w_{pg}	2.275508			
Prediction method	Qualifying (CV 4)			

A.16 SigClaimVec2

	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
number of iterations	5	5	5	5	5
η_1	7.446387E-05	0.003400406	8.276997E-06	5.207112E-05	5.96212E-06
η_2	0.03044203	0.0312788	0.02010364	0.02208705	1.835545E-06
η_3	0.03158253	0.006951657	0.03037333	0.003526475	0.0008743242
η_4	0.002342561	0.007359345	0.02837753	6.777202E-05	2.96498E-07
w_{bias}	0				
w_{sex}	3.260356				
w_{age}	0.1731932				
w_{bias1}	0.8647634				
w_{bias2}	0.9697571				
w_{bias3}	0.9358035				
$w_{provider}$	0.5224175				
w_{vendor}	1.467883				
w_{pcp}	0.1221808				
$w_{specialty}$	1.020472				
w_{place}	2.667761				
w_{los}	0.5817887				
w_{dsfs}	0.9628636				
w_{pcg}	1.831175				
w_{pg}	2.459197				
w_{isp}	0.7229536				
w_{ia}	0.4556737				
w_{is}	1.119802				
w_{ipl}	1.74992				
Prediction method	Qualifying (CV 4)				

A.17 SigClaimVec3

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_1	8.648005E-05	0.002775013	3.945398E-06	4.007937E-05
η_2	0.03044203	0.03394	0.01884128	0.02271162
η_3	0.03081222	0.009026892	0.02623151	0.003617873
η_4	0.002342561	0.007313635	0.02750883	4.740497E-05
w_{bias}	0			
w_{sex}	3.260356			
w_{age}	0.1731932			
w_{bias1}	0.8647634			
w_{bias2}	0.9697571			
w_{bias3}	0.9358035			
$w_{provider}$	0.5224175			
w_{vendor}	1.467883			
w_{pcp}	0.1221808			
$w_{specialty}$	1.020472			
w_{place}	2.667761			
w_{los}	0.5817887			
w_{dsfs}	0.9628636			
w_{pcg}	1.831175			
w_{pg}	2.459197			
w_{isp}	0.7229536			
w_{ia}	0.4556737			
w_{is}	1.119802			
w_{ipl}	1.74992			
cc_{bias}	3.633074E-07			
cc_{slope}	1.005121			
cc_{min}	0.04843435			
cc_{max}	1.008892			
cc_a	0.997853			
cc_b	0.02073831			
cc_c	0.4973778			
Prediction method	Qualifying (CV 4)			

A.18 SigClaimVec4

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_1	8.648005E-05	0.002775013	3.945398E-06	4.007937E-05
η_2	0.03044203	0.03394	0.01884128	0.02271162
η_3	0.03081222	0.009026892	0.02623151	0.003617873
η_4	0.002342561	0.007313635	0.02750883	4.740497E-05
η_5	0.001453925			
w_{bias}	1E-05			
w_{sex}	2.222178			
w_{age}	0.1390827			
w_{sexage}	0.09876291			
w_{bias1}	0.8647634			
w_{bias2}	0.9697571			
w_{bias3}	0.9358035			
$w_{provider}$	0.4701757			
w_{vendor}	1.321095			
w_{pcp}	0.1099627			
$w_{specialty}$	1.020472			
w_{place}	2.667761			
w_{los}	0.5817887			
w_{dsfs}	0.9628636			
w_{pcg}	1.648057			
w_{pg}	2.459197			
w_{isp}	0.7229536			
w_{ia}	0.3079063			
w_{is}	1.161035			
w_{ipl}	1.74992			
w_{ispl}	7.390157E-06			
w_{iapl}	8.737499E-06			
cc_{bias}	1.617096E-06			
cc_{slope}	1.030841			
cc_{min}	0.05591653			
cc_{max}	1.038396			
cc_a	1.023878			
cc_b	1.123463E-06			
cc_c	0.3876235			
Prediction method	Qualifying (70%)			

A.19 SigClaimVec5

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_1	6.658964E-05	0.002934576	3.775358E-07	6.487648E-05
η_2	0.03044203	0.03589155	0.02072541	0.02271162
η_3	0.03081222	0.008124202	0.02360836	0.00397966
η_4	0.002342561	0.007313635	0.02750883	5.830811E-05
η_5	0.001788328			
w_{bias}	1E-05			
w_{sex}	2.222178			
w_{age}	0.1390827			
w_{sexage}	0.09876291			
w_{bias1}	0.8647634			
w_{bias2}	0.9697571			
w_{bias3}	0.9358035			
$w_{provider}$	0.4701757			
w_{vendor}	1.321095			
w_{pcp}	0.1099627			
$w_{specialty}$	1.020472			
w_{place}	2.667761			
w_{los}	0.5817887			
w_{dsfs}	0.9628636			
w_{pcg}	1.648057			
w_{pg}	2.459197			
w_{isp}	0.7229536			
w_{ia}	0.3079063			
w_{is}	1.161035			
w_{ipl}	1.74992			
w_{isexpl}	7.390157E-06			
w_{iagepl}	8.737499E-06			
cc_{bias}	1.617096E-06			
cc_{slope}	1.030841			
cc_{min}	0.05591653			
cc_{max}	1.038396			
cc_a	1.023878			
cc_b	1.123463E-06			
cc_c	0.3876235			
Prediction method	Qualifying (CV 4)			

A.20 SigClaimVec6

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_1	6.658964E-05	0.002934576	3.775358E-07	6.487648E-05
η_2	0.03044203	0.03589155	0.02072541	0.02271162
η_3	0.03081222	0.008124202	0.02360836	0.00397966
η_4	0.002342561	0.007313635	0.02750883	5.830811E-05
η_5	0.001788328			
η_6	0.0009236438			
w_{bias}	1E-05			
w_{sex}	2.222178			
w_{age}	0.1390827			
w_{sexage}	0.09876291			
$w_{truncated}$	0.5125			
w_{bias1}	0.8647634			
w_{bias2}	0.9697571			
w_{bias3}	0.9358035			
$w_{provider}$	0.4701757			
w_{vendor}	1.321095			
w_{pcp}	0.1099627			
$w_{specialty}$	1.020472			
w_{place}	2.667761			
w_{los}	0.5817887			
w_{dsfs}	0.9628636			
w_{pcg}	1.648057			
w_{pg}	2.459197			
w_{isp}	0.7229536			
w_{ia}	0.3079063			
w_{is}	1.161035			
w_{ipl}	1.74992			
w_{ispl}	7.390157E-06			
w_{iapl}	8.737499E-06			
cc_{bias}	1.617096E-06			
cc_{slope}	1.030841			
cc_{min}	0.05591653			
cc_{max}	1.038396			
cc_a	1.023878			
cc_b	1.123463E-06			
cc_c	0.3876235			
Prediction method	Qualifying (CV 4)			

A.21 SigClaimVec7

	Phase 1	Phase 2	Phase 3	Phase 4
number of iterations	5	5	5	5
η_1	7.507981E-05	0.002970341	1.163757E-06	6.278235E-05
η_2	0.03044203	0.03678884	0.02971972	0.0206776
η_3	0.03237209	0.004879199	0.03153681	0.001866457
η_4	0.002342561	0.007496476	0.02959606	3.110381E-05
η_5	0.001421447			
η_6	0.0009120983			
η_7	0	1.038187E-05	1.445651E-05	1.247214E-06
w_{bias}	1E-05			
w_{sex}	2.222178			
w_{age}	0.1390827			
w_{sexage}	0.09876291			
$w_{truncated}$	0.7169875			
w_{bias1}	0.8647634			
w_{bias2}	0.8726583			
w_{bias3}	0.9358035			
$w_{provider}$	0.2825756			
w_{vendor}	1.321095			
w_{pcp}	0.120959			
$w_{specialty}$	1.255181			
w_{place}	2.667761			
w_{los}	0.5817887			
$w_{charlson}$	0.08875125			
w_{dsfs}	1.184322			
w_{pcg}	1.648057			
w_{pg}	2.459197			
w_{suplos}	0.08986406			
w_{isp}	0.7229536			
w_{ia}	0.3079063			
w_{is}	1.161035			
w_{ipl}	1.74992			
w_{ispl}	6.651141E-06			
w_{iapl}	7.863749E-06			
w_{lab}	0.9524652			
w_{drug}	1.763486			
cc_{bias}	1.499392E-06			
cc_{slope}	1.052047			
cc_{min}	0.05411786			
cc_{max}	1.040149			
cc_a	1.028764			
cc_b	8.419494E-07			
cc_c	0.3670921			
Prediction method	Qualifying(CV 10)			