# Our milestone 2 solution of the Heritage Health Prize

March 4th, 2012

Edward de Grijs Willem Mestrom

## 1 Introduction

In this paper we describe our milestone 2 solution to the Heritage Health Prize with a RMSLE score of 0.4553 on the public leaderboard. The solution is a blend of 27 models, 13 of them were already used in the milestone 1 solution of Willem Mestrom. The other 14 models include one very strong model described in chapter 2 and 3, one GBM described in chapter 4 and some very simple 'optimized constant value' models described in chapter 5. Finally the final blend is given in chapter 6.

## 2 Ensemble of Trees model

The ensemble of trees model is only used as an input (residual) for the Stochastic Gradient Descent algorithm in the next chapter. This model has been implemented from scratch.

### 2.1 Data

For this model the "Two years claims" method is used, using only release3 data. For the claims data the number of times a category occurs within the previous year is used:

Training Y2data-> Y3pred
Prediction Y3data-> Y4pred

But also the number of times a category occurs within the previous two years is used:

Training Y1data+Y2data-> Y3pred
Prediction Y2data+Y3data-> Y4pred

For the test set 25% of Y3pred is used.

### 2.2 Features

For each UserID,Year combination (339000 rows in total) a total of 175 columns are derived from the input data, as presented in the next table. The following abbreviations are used:

Age = AgeAtFirstClaim
DiH = DaysInHospital
Trunc = ClaimsTruncated
Charl = CharlsonIndex
LoS = LengthOfStay
Place = PlaceSvc
Spec = Specialty
PCG = PrimaryConditionGroup

| File | Input field | Number of categories | Conversion method | Number of resulting categories | Number of columns Y2d->Y3p | Number of columns Y1d+Y2d->Y3p |
|---|---|---|---|---|---|---|
| Members | Age | 10 | None | 10 | 1 | |
| | Age | 10 | Cluster | 3 | 1 | |
| | Sex | 3 | None | 3 | 1 | |
| DaysInHospital | DiH | 16 | None | 16 | 1 | |
| | Trunc | 2 | None | 2 | 1 | |
| Claims | Charl | 4 | Cat2Col | 4 | 4 | 4 |
| | LoS | 11 | Cat2Col | 11 | 11 | 11 |
| | Place | 9 | Cat2Col | 9 | 9 | 9 |
| | Spec | 13 | Cat2Col | 13 | 13 | 13 |
| | PCG | 46 | Cat2Col | 46 | 46 | 46 |
| | Vendor | 6388 | Cluster | 3 | 1 | 1 |
| | PCP | 1360 | Cluster | 3 | 1 | 1 |

**Tabel 1: Ensemble of trees features**

**Conversion methods**

The conversion method is used to translate the categories into columns. The following conversions can be distinguished:

**None**

Column is equal to the input column.

**Cat2Col**

Each category is translated to a separate column. Within this column the number of appearances of that category for the member,year combination is counted.

So these columns created from the claims data and converted with the Cat2Col conversion contains the number of times a certain category occurs, so for instance the maximum value for these Y2d->Y3p columns is 44. (This is the maximum number of claims in Y2data, because more claims are truncated. The best results for the ensemble of trees was obtained by translating the “number of times” values to only three values for each column: (0, 1 or 2). The probable explanation for this is that there is more information between small number of times a category occurs (for instance between 0, 1, 2) than for higher values (for instance 42, 43, 44).

The translation of “number of times” values to only three values are performed by histogram equalization, which tries to optimize the number of times that a resulting category occurs (0, 1 or 2), in such a way that they occur about equally. This is performed by placing the two thresholds between 0 and 1, and between 1 and 2, so that the values 0, 1 and 2 occur about equally times. This is performed for every column created from the claims with the Cat2Col conversion, separately. Only the resulting columns are the input for the ensemble of trees.

**Cluster**

All the categories of the last claim of each Customer,year are clustered into three different clusters. The three resulting clusters are:

1) Chi_Yates > 10.0 and CatMean > Expect+0.01
2) Chi_Yates > 10.0 and CatMean < Expect-0.01
3) All categories not falling under point 1) or 2)

Where:

Expect = the expected result (average over all)
CatMean = the mean of the category.

For a Chi_Yates description, see: http://en.wikipedia.org/wiki/Yates_correction. The three resulting clusters are treated as three categories within a single column. To calculate the Chi_Yates, Expect and CatMean values, all the used DIH values are translated to two possible values: 0 (DIH=0) or 1 (DIH>0).

## 2.3 Training

For the training a test set, 25% of the lines of Y3pred, is used. The ensemble of trees is performed in DiH. The settings used are:

Number of trees: 2000
Number of columns randomly selected for each tree: 80 (out of 175)
Percentage randomly selected rows for each tree: 12.5 %
Minimal number of observations in terminal node: 2

In the ensemble of trees, all columns are treated as categorical, except the AgeAtFirstClaim column with 10 categories, and the DaysInHospital with 16 categories, which are treated as a numerical.

## 2.4 Post Processing

The results (DiH based) are corrected by the following post-processing step, using six parameters (P1, P2, P3, P4, Pmin and Pmax):

*pred2=max(min(P1*pred+P2, Pmin), Pmax)*
*If (pred2>0.25) corrected=0.25*(4*pred2)^P3*
*else corrected=0.25*(4*pred2)^P4*

The parameters are optimized using a Genetic Algorithm for optimization.

# 3 Stochastic Gradient Descent Model

This model has been stepwise developed to obtain a combination of models that is very strong by itself. It is arguable whether this model can be called a single model. The resulting public leaderboard score is quite good (0.4585).

Often many models are used and combined by blending, but here another approach is used: the models are all combined during optimization. The advantage is that various differences in models are optimised in one step. A disadvantage is the higher chance of overfitting if the optimization is guided too much by the results of the public leaderboard score.

The stochastic gradient descent model uses the ensemble of trees model as input. This model has been implemented from scratch. The data input of this model is varied over 4 data (year) combinations (see next paragraph).

## 3.1 Data

For this model the "Two years claims" method is used, using only release3 data. For the claims data, 4 different data (year) combinations are used as input:

| **Combination** | **Used year data** | **TreeEnsemble input** |
|---|---|---|
| 1 | Training: Y2data-> Y3pred<br>Prediction: Y3data-> Y4pred | No |
| 2 | Training: Y1data+Y2data-> Y3pred<br>Prediction: Y2data+Y3data-> Y4pred | No |
| 3 | Training: max(Y1data-Y2data, 0) -> Y3pred<br>Prediction: max(Y2data-Y3data, 0) -> Y4pred | Yes |
| 4 | Training: Y1data->Y2pred, Y2data-> Y3pred<br>Prediction: Y2data->Y3pred, Y3data-> Y4pred | Yes |

For each of these combinations a separate SGD has been used, but with the same features (see next paragraph). Combinations 1 and 2 are processed without residual input, combinations 3 and 4 with residual input of the ensemble of trees model. For the test set 25% of Y3pred is used, which is equal to the test set of the ensemble of trees model.

## 3.2 Features

For this model, for each UserID,Year combination (339000 rows in total) a large number of columns are derived from the input data, as presented in the next table. The following abbreviations are used:

Age = AgeAtFirstClaim
DiH = DaysInHospital
Trunc = ClaimsTruncated
Charl = CharlsonIndex
LoS = LengthOfStay
PG = ProcedureGroup
Place = PlaceSvc
Spec = Specialty
PCG = PrimaryConditionGroup
DC = DrugCount
LC = LabCount

| Parts | Input Field(s) | Number of categories | Conversion method | Number of resulting categories | Number of columns Y2d->Y3p | Usage (see paragraph training) |
|---|---|---|---|---|---|---|
| Ensemble of Trees result | (Residual predictions) | NA | NA | NA | 1 | Directly |
| Part 1 (categories) | Age | 10 | None | 10 | 1 | Category |
| | Sex | 3 | None | 3 | 1 | Category |
| | DiH | 16 | None | 16 | 1 | Category |
| | Trunc | 2 | None | 2 | 1 | Category |
| Part 2 (Number) | Charl | 4 | Cat2Col | 4 | 4 | Number |
| | Charl | 4 | Cat2Num1 | NA | 3 | Number |
| | PCG | 46 | Cat2Col | 46 | 46 | Number |
| | PCP | 1360 | Cat2Col100 | 100 | 100 | Number |
| | LoS | 11 | Cat2Num2 | NA | 4 | Number |
| | DSFS | 13 | Cat2Col | 13 | 13 | Number |
| | PG | 18 | Cat2Col | 18 | 18 | Number |
| | Sex and Age | 3*10 | Cat2Col | 30 | 30 | Number |
| | Sex and Place | 3*9 | Cat2Col | 27 | 27 | Number |
| | Spec and Place | 13*9 | Cat2Col | 117 | 117 | Number |
| | Spec and LoS | 13*11 | Cat2Col | 143 | 143 | Number |
| | Sex and Age<T and PCG=Pregnancy or otherwise | 94 | Select1 | 94 | 94 | Number |
| | Sex and Age<T and PCG=AMI or otherwise | 34 | Select2 | 34 | 34 | Number |
| | Sex and Age<T and PCG=MSC2a3 or otherwise | 24 | Select3 | 24 | 24 | Number |
| | DC | NA | Cat2Num3 | 7 | 7 | Number |
| | LC | NA | Cat2Num3 | 7 | 7 | Number |
| Part 3 (categories) | DSFS | 13 | Combine1 | 4195 | 1 | Category |
| | PG | 12 | Combine1 | 3143 | 1 | Category |
| | Vendor | 6389 | Direct6 | 6389 | 6 | Category |
| | PCP | 1012 | Direct6 | 1361 | 6 | Category |
| | Spec, Place and PCG | 13*9*46 | Direct6 | 2160 | 6 | Category |
| | PCG, Charl and PG | 46*4*18 | Direct6 | 1951 | 6 | Category |
| | Age, PCG, Charl, PG | 10*46*18 | Direct6 | 11586 | 6 | Category |
| | DSFS, PCG, Charl, PG | 13*46*4*12 | Direct6 | 3494 | 6 | Category |
| | LoS, PCG, Charl, | 11*46*4*12 | Direct6 | 12912 | 6 | Category |

| Parts | Input Field(s) | Number of categories | Conversion method | Number of resulting categories | Number of columns Y2d->Y3p | Usage (see paragraph training) |
|---|---|---|---|---|---|---|
| | PG | | | | | |
| | Age, DiH, Trunc | 10*16*2 | Direct6 | 436 | 6 | Category |
| | DSFS, Trunc | 13*2 | Direct6 | 26 | 6 | Category |
| | Age, DiH | 10*16 | Direct6 | 152 | 6 | Category |
| | Trunc, PayDelay, SupLOS | 2*?*2 | Direct6 | 584 | 6 | Category |
| | Charl | 4 | Direct6 | 4 | 6 | Category |
| | Charl, Place | 4*9 | Direct6 | 36 | 6 | Category |
| | Age, Place and min_DC<7 | 10*9*1 | Direct6 | 90 | 6 | Category |
| | PCG and last_LC<8 | 46*1 | Direct6 | 46 | 6 | Category |

**Tabel 2: Stochastic gradient descent features**

**Conversion methods**

The conversion method is used to translate the categories into columns. The following conversions can be distinguished:

**None**

Column is equal to the input Column.

**Cat2Col**

Each category (or category combination) is translated to a separate column.

Within this column the number of appearances of that category for the member,year combination is counted.

**Cat2Col100**

The most occurring 100 categories are chosen. Each of these 100 categories is translated to a separate column. Within this column the number of appearances of that category for the member,year combination is counted.

**Cat2Num1**

Categories of Charlson are translated to numbers. For each member,year three new columns are calculated, containing the sum, maximum and average for that member,year combination. (Used string to number conversion for Charl: “0”=>0, “1-2”=>3, “3-4”=>7, “5+”=>10)

**Cat2Num2**

Categories of LengthOfStay are translated to number of days. For each member,year four new columns are calculated, containing the sum, maximum, number of claims > 0 days and average for that member,year combination.

(Used string to number conversion for weeks: “1-2weeks”=10, ”2-4weeks”=21, “4-8weeks”=42, “8-12weeks”=70, “12-26weeks”=120, “26+weeks”=200)

**Cat2Num3**

The numbers of DrugCount and LabCount are converted:
For each member,year seven new columns are calculated, containing the number of lines, sum, minimum, maximum, last claim number, variation and average for that member,year combination. The used variation is calculated by the sum of the absolute differences between the lines of the member,year combination.

**Select1**

With Select1 new categories are derived which consists of

1) PCG=Pregnancy or otherwise
2) Age<T where T is an age Threshold, to cluster all the ages under the Threshold on one category, and all the ages above or equal to the Threshold onto another category. The following thresholds are used: 10, 20, 30, 40, 50, 60, 70, 80
3) Sex.

At most 2*2(*8)*3 = 96 distinctive categories could be derived, but 2 of them were nonexistent, so in total 94 distinctive categories remain.

Each category combination is translated to a separate column. Within this column the number of appearances of that category for the member,year combination is counted.

**Select2**

With Select2 new categories are derived which consists of

1) PCG=AMI or otherwise
2) Age<T where T is an age Threshold, to cluster all the ages under the Threshold on one category, and all the ages above or equal to the Threshold on another category. The following thresholds are used: 20, 40, 70
3) Sex.

At most 2*2(*3)*3 = 36 distinctive categories could be derived, but 2 of them were nonexistent, so in total 34 distinctive categories remain.

Each category combination is translated to a separate column. Within this column the number of appearances of that category for the member,year combination is counted.

**Select3**

With Select3 new categories are derived which consists of

1) PCG=MSC2a3 or otherwise
2) Age<T where T is an age Treshold, to cluster all the ages under the Threshold on one category, and all the ages above or equal to the Threshold on another category. The following thresholds are used: 40, 70
3) Sex.

At most 2*2(*2)*3 = 24 distinctive categories are derived.

Each category combination is translated to a separate column. Within this column the number of appearances of that category for the member,year combination is counted.

**Combine1**

With Combine1 combinations are made within the input fields (DSFS or ProcedureGroup), to combine different categories for different claims into one result for a member,year combination. Because there are many possible combinations of categories, there will be many resulting combinations. These combinations will be treated as “new” categories in the resulting output column.

**Direct6**

With Direct6 the categories from the last six claim rows of each member,year combination are used to create the resulting six columns. If the input field consists of a combination of data input columns, all the possible combinations are observed as different categories. Because not all combinations are present in the input data, the number of resulting categories is lower than the total number of possible combinations.

All the calculated feature columns are used in the training and predicting algorithm.

## 3.3 Training

For the training, a test set, 25% of the lines of Y3pred is used, the same test set that has been used for the Ensemble of trees algorithm. There are four training + prediction sequences, equal to the four combinations as described in the DATA paragraph. Only for combinations 3 and 4 the Ensemble of Trees model has been used as an extra input column. The Stochastic Gradient Descent is performed in DiH without mini-batches. For each parameter to learn there is a learning rate η and a shrinkage parameter λ. For each training case (member,year combination), all applicable parameters are updated using the update rule: (performed in this order)

weight = weight + η *gradient
weight = weight * (1 – η * λ)

Every iteration over de dataset is performed identically, from the lowest member ID to the highest member ID. Trainings stops as soon at the test set score starts to increase, but only a maximum of 200 iterations is permitted.

After the first optimization phase, which stops due to an increase of the test set score, another six phases are used. Before every successive phase, all the weights are reduced by multiplying them by 0.9, and the global learning rates are halved. Each of these six phases also stops when the test set score starts to increase. The weights belonging to the best (=lowest) found test set score after every iteration are used as the prediction weights.

All weight parameters are initialized to zero, and are learned simultaneously.

The global learning rates and various other parameters are optimized using a simple stepwise change for each separate parameter. The following paragraphs will also explain these various other parameters.

## 3.4 Handling Ensemble of trees input

This input consists of predictions that were calculated in the Ensemble of Trees model. The weight for this input is calculated at the start, and is not changed thereafter in the training phase. This weight is calculated at the start by optimizing the RMSE result, where only this column is used.

## 3.5 Used weight Bias

The weight bias used for every member,year data row is 0.02 if the Ensemble of Trees input is used and 0.44 otherwise. The lower bias value is caused by the Ensemble of Trees input which is used directly (and not as residual, where the mean value is eliminated).

## 3.6 Intermediate processing

The intermediate processing is the same as the post processing of the Ensemble of Trees method, but now this processing is used during the processing after every ten iterations, because this processing influences the test set result.

It is the (increasing) test set result that stops the iterations, so the test set result is often updated.

## 3.7 Usage of numbered items (part 3)

In Table 2, under the column "Usage" the categories are all learned with one weight for each category. The rows with a "Number" Usage are converted to separate items in the following manner:

1) The number divided by the maximum value of that column.
   (This gives a value in between 0.0 and 1.0) (global learning rate1)
2) An extra weight if the number is higher than the Thresholds:
   0,1,2,3,4,5,6,7,8,10,12,14,16,19,22,25 or 29.
   (This gives 17 extra distinctive items, each with a separate weight.)
   (Threshold=0 with global learning rate 2)
   (Threshold=1 with global learning rate 3)
   (All thresholds>1 with global learning rate 4)

The rationale of step 2 is that a certain (number) value can be indicative of a certain severity of a certain number of claims of this kind. The used method makes a broad spectrum of thresholds with separate weights, which can now be used to be optimized using Stochastic Gradient Descent. Note that there are more thresholds at lower values, because those lower values are more important. Using higher thresholds than 29 was not tested beneficial.

## 3.8 Learning rates

There are global, intermediate and local learning rates factor variables that in total will make up the learning rate of a specific item. The global learning rates usage in relation with the thresholds are denoted in the previous paragraph "Usage of numbered items".

These global learning rates are

| Global learning rates | Used for part | Initial without TreeEnsemble input | Initial with TreeEnsemble input |
|---|---|---|---|
| lrate1 | 1 and 2 | 0.0001660 | 0.0001294 |
| lrate2 | 1 and 2 | 0.0000080 | 0.0000026 |
| lrate3 | 1 and 2 | 0.0000050 | 0.0000013 |
| lrate4 | 1 and 2 | 0.0000030 | 0.0000010 |
| lrate | 3 only | 0.0000330 | 0.0000259 |

**Tabel 3: Global learning rates**

The global learning rates are all multiplied by 0.7 after 10 iterations and again after 20 iteration. Often a larger learning rate in the beginning of a SGD is beneficial. After each phase all the global learning rates are halved.

The intermediate learning rates factors are only used for the part 3 features. In the following table, all these intermediate learning rate factors are summarized. Note that the larger importance of last claim within each member,year combination is often the most important one (has largest learning rate factor). The intermediate learning rate factors for part 1 and 2 are not used (they are all 1.00).

| **Parts** | **Input Field(s)** | **# columns Y2d->Y3p** | **Learnrate factor 2, used column** | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | **Last** | **Last-1** | **Last-2** | **Last-3** | **Last-4** | **Last-5** |
| Part 3 (categories) | DSFS | 1 | 0.20 | | | | | |
| | PG | 1 | 0.14 | | | | | |
| | Vendor | 6 | 0.14 | 0.67 | 0.13 | 0.10 | 0.30 | 0.09 |
| | PCP | 6 | 2.00 | 2.97 | 2.17 | 1.00 | 0.94 | 0.69 |
| | Spec, Place and PCG | 6 | 32.97 | 3.33 | 5.33 | 4.00 | 2.67 | 0.82 |
| | PCG, Charl and PG | 6 | 1.20 | 4,56 | 1.30 | 0.60 | 0.71 | 0.20 |
| | Age, PCG, Charl, PG | 6 | 0.40 | 0.12 | 0.06 | 0.02 | 0.08 | 0.01 |
| | DSFS, PCG, Charl, PG | 6 | 0.09 | 0.19 | 0.27 | 0.20 | 0.24 | 0.07 |
| | LoS, PCG, Charl, PG | 6 | 0.24 | 0.10 | 0.08 | 0.03 | 0.05 | 0.01 |
| | Age, DiH, Trunc | 6 | 1.60 | 0.82 | 0.47 | 0.57 | 0.38 | 0.11 |
| | DSFS, Trunc | 6 | 5.20 | 0.95 | 0.18 | 0.80 | 0.42 | 0.27 |
| | Age, DiH | 6 | 1.60 | 1.05 | 0.84 | 0.63 | 0.26 | 0.16 |
| | Trunc, PayDelay, SupLOS | 6 | 5.70 | 0.67 | 0.33 | 0.25 | 0.33 | 0.34 |
| | Charl | 6 | 2.85 | 1.33 | 0.37 | 1.60 | 0.33 | 0.87 |
| | Charl, Place | 6 | 3.30 | 0.52 | 0.33 | 0.40 | 1.07 | 0.16 |
| | Age, Place and min_DC<7 | 6 | 1.35 | 2.17 | 0.44 | 1.01 | 0.53 | 0.27 |
| | PCG and last_LC<8 | 6 | 0.76 | 0.97 | 0.90 | 1.12 | 0.66 | 0.13 |

**Tabel 4: Intermediate learning rates**

There are also local learning rates for every “Number” row in the “Usage” column of table 2. This seems very specific, because there are tens of thousands of weights here, but the learning rate correction depends on the number of times a certain (weight able) item occurs in the dataset. The used (hand tuned) formula used is:

local_ learnrate_factor = 2.0/(0.3 * log( 0.05*(counted_items+1.0 ))  (log=natural logarithm)

The total learning rate is the global learnrate multiplied by the intermediate learning rate factor, and this local_learnrate_factor.

With this formula the learning rate of items occurring only a few times are more than ten times as large as the learning rate of items that occur a thousand times or more. It seems counter intuitive to favor items that occur only a few times, but it is beneficial for the current data set result.

Furthermore the local learning rates are multiplied by 4 if the maximum of the column is 1, and they are multiplied by 2 if the maximum of the column is 2.

## 3.9 Regularisation

Only for part 3 (see table 2), a regularization is used with a value of λ=0.50. For parts 1 and 2 no regularisation is used (λ=0.00).

## 3.10 Gradient correction

A gradient correction is used which improved the results. This correction corrects the difference between every DiH value, and prediction value. It could well be that this correction is unnecessary if log(DiH+1) was used (instead of plain DiH) values for calculating the predictions.

This gradient is always corrected by means of:

*if (gradient<-1.0)* *gradient=-((-gradient)^1.776)*
*if (gradient>-1.0 and gradient<0.0)* *gradient=-((-gradient)^1.600)*
*if (gradient>0.0 and gradient<1.0)* *gradient=gradient^0.947*
*if (gradient>1.0)* *gradient=gradient^0.343*

The four (exponent) parameters were optimized using a simple stepwise optimization.

## 3.11 Blending the four combinations

The results of the four combinations are calculated using a testset of 25% of the Y2data as a training set. (The same testset as used for the Ensemble of Trees input). The combinations are combined by linear blending of the testset results. The result is denoted as model c279 in chapter 5, and resulted in a public leaderboard score of 0.4585.

# 4 GBM

Seeing the nice results the Market Makers got using GBM we decided to give it a try as well. Since it is not our style to take a standard implementation we build our own. This not only helps to learn and understand the algorithm it also gives opportunities to tune the algorithm for the HHP dataset.

## 4.1 Basic algorithm

The basic GBM algorithm is fairly simple:

1. Start with the overall mean as predictor
2. Fit a simple base predictor to the residuals
3. Update the predictor and residuals by adding the predictor from step 2 multiplied by a small stepsize
4. Repeat from step 2 until some stopping criteria.

For a more thourough description see "Greedy Function Approximation: A Gradient Boosting Machine"[1] or the wikipedia article "Gradient boosting"[2]. Many different settings were tried but only one GBM made it into our final blend. The base predictor used for this model is a regression tree with a maximum of 4 splits and at least 50 cases in each leave node. The model consist of 2800 trees (iterations), the stepsize is set to 0.05 and each tree is build using a distinct 50% random subset of the training set.

## 4.2 Features

The model is build using the "one-year-history" setup as described in the milestone 1 paper from Willem Mestrom. Most but not all of the features as described by the Market Makers in their milestone 1 paper were used. For the selection of the combinations of PrimaryConditionGroup × Specialty, ProcedureGroup × Specialty, ProcedureGroup × PrimaryConditionGroup and PrimaryConditionGroup × PlaceOfService we used a different method. Instead of using logistic regression a BaggedTree predictor was build for each of the 4 combination groups. The BaggedTree predictor consisted of 1000 trees, each with a maximum depth of 30 levels and a minimum of 100 observations in each leave. The combinations that were most frequently used in the trees were selected for the GBM. This procedure was repeated multiple times and some arbitrary choices were made for combinations which would be selected by one run but not by another. For completeness the full set of features is listed here (the full list of the selected combinations can be found in appendix A).

| Variable | Number of columns | Description |
|---|---|---|
| Age | 10 | 0/1 |
| Sex | 3 | 0/1 |
| NoClaims | 1 | Count |
| ClaimsTruncated | 1 | 0/1 |
| Specialty | 13 | 0/1 |
| PlaceSvc | 9 | 0/1 |
| PrimaryConditionGroup | 46 | 0/1 |

[1] http://www-stat.stanford.edu/~jhf/ftp/trebst.pdf
[2] http://en.wikipedia.org/wiki/Gradient_boosting

| Variable | Number of columns | Description |
|---|---|---|
| ProcedureGroup | 18 | 0/1 |
| SupLOS | 3 | 0/1 |
| LengthOfStay UNKNOWN | 1 | 0/1 |
| LengthOfStay KNOWN | 1 | 0/1 |
| Distinct ProviderId | 1 | Count |
| Distinct Vendor | 1 | Count |
| Distinct PCP | 1 | Count |
| Distinct Specialty | 1 | Count |
| Distinct PlaceSvc | 1 | Count |
| Distinct PrimaryConditionGroup | 1 | Count |
| Distinct CharlsonIndex | 1 | Count |
| Distinct ProcedureGroup | 1 | Count |
| Distinct SupLOS | 1 | Count |
| LengthOfStay min | 1 | Number |
| LengthOfStay max | 1 | Number |
| LengthOfStay avg | 1 | Number |
| LengthOfStay range | 1 | Number |
| LengthOfStay std | 1 | Number |
| DSFS min | 1 | Number |
| DSFS max | 1 | Number |
| DSFS avg | 1 | Number |
| DSFS range | 1 | Number |
| DSFS std | 1 | Number |
| CharlsonIndex min | 1 | Number |
| CharlsonIndex max | 1 | Number |
| CharlsonIndex avg | 1 | Number |
| CharlsonIndex range | 1 | Number |
| CharlsonIndex std | 1 | Number |
| NoDrugs | 1 | Count |
| Drugs min | 1 | Number |
| Drugs max | 1 | Number |
| Drugs avg | 1 | Number |
| Drugs range | 1 | Number |
| Drugs std | 1 | Number |
| NoLabs | 1 | Count |
| Labs min | 1 | Number |
| Labs max | 1 | Number |
| Labs avg | 1 | Number |
| Labs range | 1 | Number |
| Labs std | 1 | Number |
| PrimaryConditionGroup × Specialty | 47 | Count, see appendix A |
| ProcedureGroup × Specialty | 42 | Count, see appendix A |
| ProcedureGroup × PrimaryConditionGroup | 35 | Count, see appendix A |
| PrimaryConditionGroup × Place | 51 | Count, see appendix A |

**Tabel 5: GBM features**

## 4.3 Performance optimizations

To obtain good results with a GBM a lot of iterations are needed. To speed things up we implemented some optimizations which may affect the results. First a simple optimization is rounding the inputs from floating point numbers to single byte integers. This can speed things up considerably if your system has (relatively) low memory bandwidth. In this dataset it has very little impact on the result since most inputs are integer numbers between 0 and 45. A second optimization is to use a dynamic step size. The reason so many iterations are needed is because the results get better with smaller stepsizes. To allow for somewhat lager steps we reduced the stepsize for predictors that don't generalize well. Each base predictor is build using only half the training data. Predictions are then made for the held out part of the training data. A scaling factor $\alpha$ is calculated to get the optimal fit on the held out part. The stepsize for this step is now multiplied by $\alpha^2$. This way the stepsize gets smaller when a base predictor doesn't generalize well. The overall result is that equally good results can be obtained using less iterations.

# 5 Optimizing constant values

It is known that the overall leaderboard average differs from other years, and could be corrected. Investigations show that there are more mean differences between the last year (Y4pred) in relation to the years before that. This can have many causes, for instance the data has been gathered in a (slightly) different way, or the used questionnaire has been changed, etc. The leaderboard score shows some insight into these differences.

The differences should be very small, but there are differences that are so large that there must be a cause for this. Clearly we cannot find the real cause because we have no insight in the data gathering methods. For instance a large difference has been found of the sex=undefined situation between the last year, and the years before. An undefined sex could be caused by not filling in a certain questionnaire, or by another cause, but it is very typical, because this undefined category has a much higher chance on hospitalization than sex=female or sex=male. This difference led to the investigation of other differences, which are present in the dataset.

The following significant differences were found:

| | **Average over** |
|---|---|
| all mean | All rows |
| m1 | Number of Claims>0 in Y2 |
| m2 | Number of Claims>0 in Y1 and Y2 |
| m3 | Sex = Undefined |
| m4 | DaysInHospital in Y3 = 0 |
| m5 | AgeAtFirstClaim = 5 |
| m6 | Total of CharlsonIndex in Y3 = 0 |
| m7 | Number of times PlaceSvc=Office in Y3 = 0 |
| m8 | Specialty in Y3 = Internal |
| m9 | Specialty in Y3 = Diagnostic |
| m10 | Specialty in Y3 = General Practice |
| m11 | DSFS in Y3 = 3-4 months |
| m12 | DSFS in Y3= 8-9 months |

**Tabel 6: Optimized constant models**

# 6 Final blend

The final solution is a blend of 27 models, 13 of them were already used in the milestone 1 solution of Willem Mestrom. The other 14 models include one very strong model (c279) as described in chapter 2 and 3, one GBM model (GBM2) as described in chapter 4 and the 'optimized constant value' models as described in chapter 5. The weights were calculated using the same procedure as used for milestone 1.

| Model | Algorithm | RMSLE (Leaderboard) | Weight |
|---|---|---|---|
| CatVec1 | SDG | 0.4758 | 0.048 |
| CatVec2 | SDG | 0.4666 | -0.116 |
| CatVec3 | SDG | 0.4666 | -0.075 |
| SigCatVec1 | SDG | 0.4644 | 0.142 |
| SigCatVec2 | SDG | 0.4657 | -0.077 |
| PerClaim | SDG | 0.4640 | 0.104 |
| SigCatVec5 | SDG | 0.4625 | 0.194 |
| SigCatVec3c-Y3 | SDG | 0.4750 | 0.256 |
| SigCatVec3b | SDG | 0.4656 | -0.139 |
| SigCatVec7 | SDG | 0.4645 | 0.145 |
| SigCatVec6 | SDG | 0.4633 | -0.061 |
| SicClaimVec7 | SDG | 0.4606 | 0.239 |
| GBM2 | GBM | 0.4626 | 0.089 |
| c279 | TreeEnsemble + SGD | 0.4585 | 0.413 |
| all mean | average | 0.4865 | -0.057 |
| m1 | average | 0.4913 | 0.079 |
| m2 | average | 0.4904 | -0.063 |
| m3 | average | 0.4844 | -0.381 |
| m4 | average | 0.4811 | 0.123 |
| m5 | average | 0.4854 | -0.197 |
| m6 | average | 0.4833 | 0.075 |
| m7 | average | 0.4876 | -0.092 |
| m8 | average | 0.4855 | 0.096 |
| m9 | average | 0.4840 | 0.093 |
| m10 | average | 0.4886 | 0.089 |
| m11 | average | 0.4844 | -0.076 |
| m12 | average | 0.4831 | 0.128 |

**Tabel 7: Scores and weights of models in the final blend**

# Appendix A GBM selected combinations

## A.1 PrimaryConditionGroup × Specialty

MSC2a3-Laboratory
METAB3-Laboratory
ARTHSPIN-DiagnosticImaging
MSC2a3-Internal
MSC2a3-GeneralPractice
ARTHSPIN-Internal
ROAMI-Internal
NEUMENT-Surgery
METAB3-Internal
MSC2a3-DiagnosticImaging
MISCHRT-Internal
AMI-Internal
GIBLEED-Emergency
GIBLEED-DiagnosticImaging
ARTHSPIN-GeneralPractice
GIBLEED-Internal
HEART2-Internal
NEUMENT-Internal
ARTHSPIN-Surgery
METAB3-GeneralPractice
MISCHRT-GeneralPractice
RESPR4-GeneralPractice
SKNAUT-Internal
ROAMI-DiagnosticImaging
RESPR4-Internal
HEMTOL-Laboratory
RESPR4-DiagnosticImaging
NEUMENT-GeneralPractice
PRGNCY-Laboratory
ROAMI-Emergency
TRAUMA-Emergency
COPD-DiagnosticImaging
COPD-Internal
ARTHSPIN-Emergency
AMI-GeneralPractice
GIBLEED-GeneralPractice
RESPR4-Emergency
GYNEC1-ObstetricsAndGynecology
UTI-Laboratory
MISCL5-Emergency
HEART4-DiagnosticImaging
MISCL5-Internal
SKNAUT-GeneralPractice
CHF-Internal
RENAL2-Internal
SEIZURE-Internal
MSC2a3-ObstetricsAndGynecology

## A.2 ProcedureGroup × Specialty

PL-Laboratory
APPCHOL-Internal
CANCRB-DiagnosticImaging
APPCHOL-GeneralPractice
CATAST-Laboratory
APPCHOL-Emergency
ARTHSPIN-Internal
ARTHSPIN-GeneralPractice
APPCHOL-Surgery
ARTHSPIN-Emergency
ARTHSPIN-Surgery
CHF-Emergency
CHF-Internal
APPCHOL-Pediatrics
CANCRA-GeneralPractice
APPCHOL-ObstetricsAndGynecology
CHF-Surgery
CANCRB-Internal
AMI-Anesthesiology
APPCHOL-Other
FXDISLC-Internal
ARTHSPIN-Pediatrics
CANCRA-Internal
CANCRB-GeneralPractice
CANCRA-Pediatrics
GIOBSENT-Internal
ARTHSPIN-DiagnosticImaging
CANCRA-Pathology
CANCRB-Surgery
ARTHSPIN-Other
CATAST-Internal
FXDISLC-Other
FXDISLC-GeneralPractice
ARTHSPIN-Rehabilitation
CATAST-GeneralPractice
GIOBSENT-Surgery
CANCRA-Emergency
HEART2-Internal
CANCRA-ObstetricsAndGynecology
CANCRB-Emergency
COPD-Surgery
APPCHOL-Anesthesiology

## A.3 ProcedureGroup × PrimaryConditionGroup

PL-MSC2a3
APPCHOL-MSC2a3
CANCRA-METAB3
CATAST-MSC2a3
APPCHOL-ARTHSPIN
APPCHOL-MISCHRT
APPCHOL-GIBLEED
CANCRB-ARTHSPIN
ARTHSPIN-MSC2a3
APPCHOL-METAB3
APPCHOL-NEUMENT
APPCHOL-RESPR4
CATAST-METAB3
CANCRB-MSC2a3
APPCHOL-AMI
APPCHOL-SKNAUT
ARTHSPIN-NEUMENT
ARTHSPIN-ROAMI
CANCRB-GIBLEED
APPCHOL-HEART2
CANCRB-ROAMI
APPCHOL-INFEC4
ARTHSPIN-ARTHSPIN
ARTHSPIN-HEART2
APPCHOL-TRAUMA
APPCHOL-MISCL5
ARTHSPIN-AMI
APPCHOL-COPD
APPCHOL-ROAMI
APPCHOL-HEART4
APPCHOL-RENAL3
CANCRB-RESPR4
CANCRB-COPD
APPCHOL-SEIZURE
APPCHOL-CHF

## A.4 PrimaryConditionGroup × Place

MSC2a3-Office
MSC2a3-IndependentLab
ARTHSPIN-Office
METAB3-IndependentLab
NEUMENT-Office
METAB3-Office
MISCHRT-Office
RESPR4-Office
GIBLEED-Office
SKNAUT-Office
AMI-Office
GIBLEED-UrgentCare
INFEC4-Office
ROAMI-InpatientHospital
COPD-Office
HEART4-Office
HEART2-Office
ROAMI-UrgentCare
GYNEC1-Office
ARTHSPIN-OutpatientHospital
TRAUMA-Office
RENAL3-Office
MISCL5-Office
ODaBNCA-Office
PRGNCY-IndependentLab
MSC2a3-OutpatientHospital
RESPR4-UrgentCare
TRAUMA-UrgentCare
HEMTOL-IndependentLab
CHF-Office
PRGNCY-Office
MISCL5-UrgentCare
SEIZURE-Office
UTI-IndependentLab
MISCHRT-IndependentLab
ARTHSPIN-UrgentCare
ROAMI-Office
RENAL2-Office
GIBLEED-InpatientHospital
CANCRB-Office
INFEC4-UrgentCare
UTI-Office
SIS-Office
HEMTOL-Office
CANCRB-IndependentLab
GIBLEED-IndependentLab
HEART2-IndependentLab
NEUMENT-UrgentCare
SEIZURE-UrgentCare
ODaBNCA-IndependentLab
GYNEC1-IndependentLab