Using GLM to improve predictive capabilities of Random Forest on H_2O Statistical Runtime for Big Data

This is a brief description of a technique used to build a model for real world dataset collected for credit card offer conversions. Data is dense about 71 cols, messy with a few features partially populated.

We use H_2O Statistical Runtime to ingest raw data and inspect the data for missing rows. We then use Generalized Linear Modeling (GLM) to identify the most important features for predicting conversions. Random Forest produces improved accuracy (0.26% to 0.17%) by ignoring features that did not matter. This workflow resulted in an accuracy improvement of up to 900 basis points over either of the techniques by themselves. (Prediction is better in comparison with best models via SPSS and SAS done by customer)

Introduction

We describe the technique of daisy chaining as two powerful algorithms to achieve better accuracy for Data Modeling. (We use the opportunity to describe operations in H_2O that help prove.)

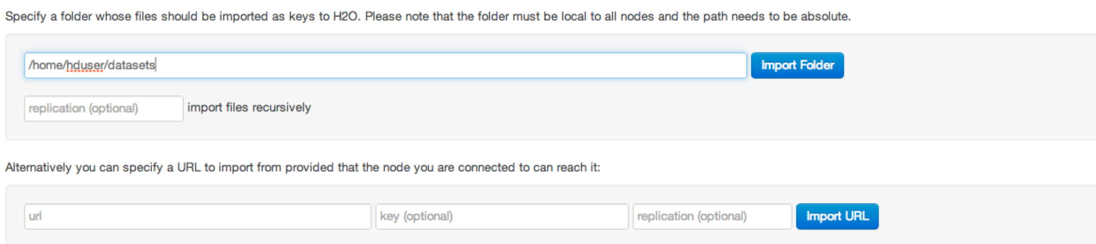
H2O is an extensible statistical runtime on top of HDFS and big data. H2O scales machine learning and modeling over large datasets. H2O gives approximate results at each stage of computation for Adhoc analysis via familiar R-like syntax and workflows. It is easy to install and integrate via JSON & REST APIs. H2O is the new wave SAS for Big Data.

Business problem:
The goal of this model is to predict "Converts" from initial dataset of user population to actual activated credit card users. Best models in this space are 77% accurate in predicting conversions. Customer used SPSS and SAS to get best models to predict.

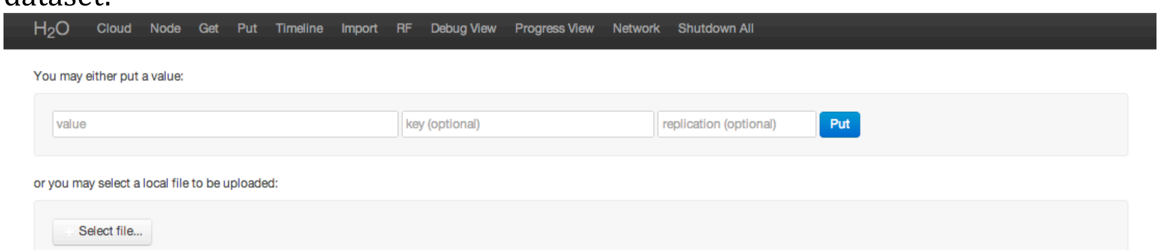
We use Out Of Bag Error estimate as presented by Breiman's paper on Random Forest in this experiment. Validation of model was also done using a separate test and training dataset.

Import dataset

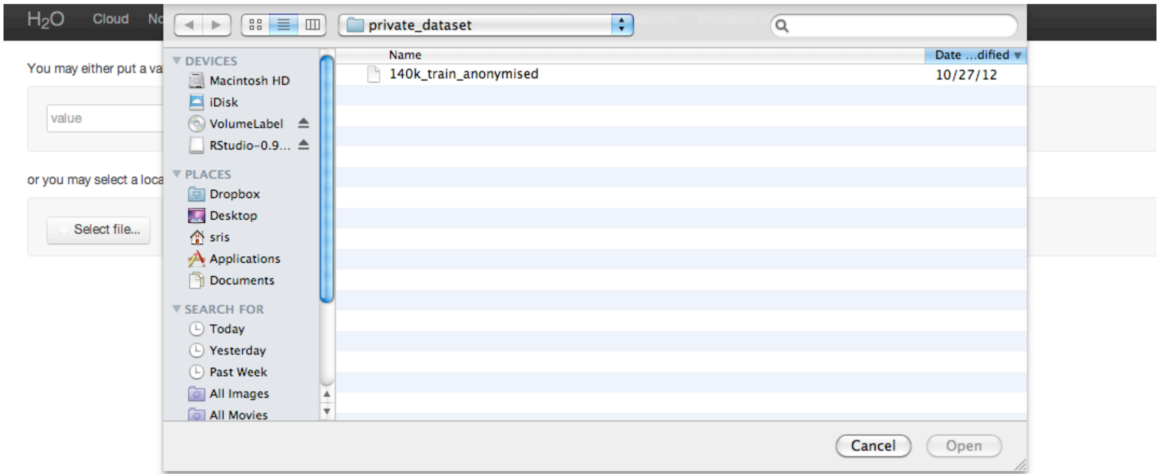
1. Import the dataset via, Import Folder
   Or clicking store view (incase of HDFS launch) HDFS files appear with hdfs://dataset/covtype.data



2. Put operation can also be used for importing the dataset:



Standard file selection interface -



3. Parse the dataset is a simple click through the link of the Key after Put or via the Store.

   This takes a few minutes for large datasets.

4. Inspect the data for a summary:

Generated from nfs://Users/sris/_work/hexbase_xem/smalldata/hhp_9_17_12.predict.data.gz_UNZIPPED by 'basic_parse'

224 Bytes-per-row * 70942 Rows = Totalsize 15891008

Parsed 217 columns

Column	0	1	2	3	4	5	6	7	8	9	10	11	12
Record offset	+0	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15
Column bytes	4b	1b	1b	1b	1b	1b	1b	1b	1b	1b	1b	1b	1b
Internal scaling	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)
Min/Max	210 - 99997488	0 - 0	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 0	0 - 1
µ	49915295.8566		0.1374	0.1543	0.1205	0.1279	0.0991	0.0839	0.0885	0.0614	0.0672	0	0.451
σ	28933883.8684		0.3443	0.3612	0.3256	0.334	0.2988	0.2772	0.284	0.24	0.2503	0	0.4976
Rows													

Column	13	14	15	16	17	18	19	20	21	22	23	24
Record offset	+16	+17	+18	+19	+20	+21	+22	+23	+24	+25	+26	+27
Column bytes	1b	1b	1b	1b	1b	1b	1b	1b	1b	1b	1b	1b
Internal scaling	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)
Min/Max	0 - 1	0 - 0	0 - 44	0 - 44	0 - 42	0 - 35	0 - 24	0 - 29	0 - 42	0 - 39	0 - 38	0 - 3
µ	0.3648	0	3.3316	3.0321	2.1727	1.0083	0.9992	0.5838	0.4783	0.4041	0.2745	0.17
σ	0.4814	0	5.0641	4.8401	3.268	2.2944	1.6401	1.3533	1.9259	1.5787	1.6654	0.88
Rows												

H_2O Cloud Node Get Put Timeline Import RF Debug View Progress View Network Shutdown All

anony.hex

Generated from anony by 'basic_parse'

179 bytes-per-row * 142982 Rows = Totalsize 24.4 MB

Parsed 71 columns

Column	Month	sex	Day_Week	TimeofDay	WebApp	age	AnsweredSurvey	Srvy_Plan2DD	Srvy_bythngs_online	Has_bnk_AC	RegisteredOnline	Population	HouseholdsPerZipCode	WhitePopulation
Record offset	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+15	+17
Column bytes	1b	1b	1b	1b	1b	1b	1b	1b	1b	1b	1b	4b	2b	4b
Internal scaling	(X+8)	(X)	(X+1)	(X)	(X)	(X+13)	(X)	(X)	(X)	(X)	(X)	(X+7)	(X)	(X)
Min/Max	8 - 11	0 - 2	1 - 7	0 - 5	0 - 1	13 - 95	0 - 1	0 - 1	0 - 1	0 - 0	0 - 1	7 - 114124	0 - 48391	0 - 86186
µ	9.4955		3.9189		0.7001	32.538	0.523	0.2859	0.0581	0	0.4126	31154.0178	12307.4286	15604.0944
σ	1.1164	0	1.8263	0	0.4582	11.2499	0.4995	0.4518	0.2339	0	0.4923	18872.1362	6873.2418	11667.8583
Rows missing data		759												
Row 0	10	M	5	16 to 18	1	47	1	0	0	0	1	21453	9825	20048
Row 1	10	F	5	16 to 18	1	43	1	0	0	0	0	14535	6384	5855
Row 2	10	F	5	16 to 18	1	21	1	0	0	0	1	23470	9469	12402

GLM

Generalized Linear Models are a powerful toolkit in any data modeler's hands.
We now run GLM using REST-API call –
Using L1 Regularization and a lambda that is high – 0.01 – we are able to detect features that do not matter.

```
http://localhost:54321/GLM?Key=anony.hex&Y=Converted&norm=L1&lambda=1e-2&family=binomial&xval=10
```

H2O Cloud Node Get Put Timeline Import RF Debug View Progress View Network Shutdown All

GLM Parameters

family: binomial **link:** logit **norm:** L1 **λ:** 0.01 **ρ:** 0.01 **α:** 1.0

Coefficients

Month	sex	Day_Week	TimeofDay	WebApp	age	AnsweredSurvey	Srvy_Plan2DD	Srvy_bythngs_online	Has_bnk_AC	RegisteredOnline	Population	HouseholdsPerZipCode	WhitePopulation	BlackPopul
0	0	0	-0.0298	-0.2517	0.0253	0	0.0189	0	0	0.6597	0	0	0	0

Model SRC

```
y = 1/(1 + Math.exp(0.0298*x[TimeofDay] + 0.2517*x[WebApp] - 0.0253*x[age] - 0.0189*x[Srvy_Plan2DD] - 0.6597*x[RegisteredOnline] + 0.072*x[CityType]
+ 0.0216*x[division] + 0.0174*x[region] - 0.0123*x[CBSAPop2003] - 0.0107*x[Innovis_pass] + 0.0211*x[checkpointscore] + 0.0249*x[grade] +
0.059*x[white_percent] + 0.5789))
```

Validation

Degrees of freedom:	141812 total (i.e. Null); 141742 Residual
Null Deviance	55990236.128
Residual Deviance	155027.292
AIC	155169.292
Training Error Rate Avg	0.2639
False Positives	0.0063
False Negative	0.2576

10 fold Cross Validation

decision threshold = %threshold

	Mean	Variance
Error rate	0.2639	
True Positive	0.6794	0.0307
True Negative	0.0106	0.0001
False Negative	0.2444	0.004
False Positive	0.0061	

Individual Models

Model 1

	Y_{real}=0	Y_{real}=1
Y_{model}=0	10065	3722
Y_{model}=1	85	300

Table1: GLM with L1 Regularization produces a list of features that might not matter for predicting converted

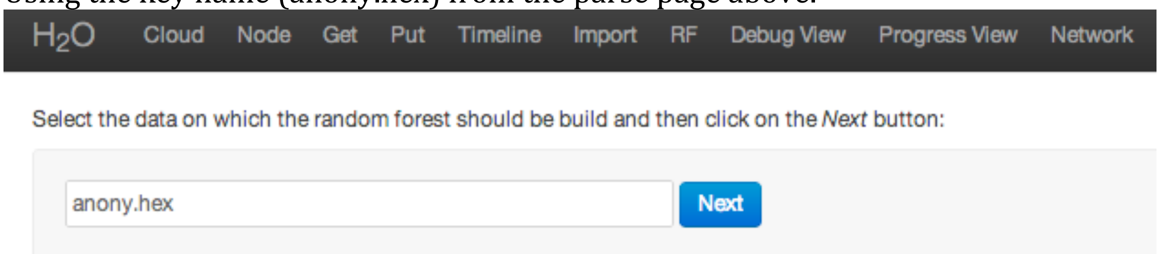
Model SRC

```
y = 1/(1 + Math.exp(0.0298*x[TimeofDay] + 0.2517*x[WebApp] - 0.0253*x[age] -
0.0189*x[Srvy_Plan2DD] - 0.6597*x[RegisteredOnline] + 0.072*x[CityType] +
0.0216*x[division] + 0.0174*x[region] - 0.0123*x[CBSAPop2003] -
0.0107*x[Innovis_pass] + 0.0211*x[checkpointscore] + 0.0249*x[grade] +
0.059*x[white_percent] + 0.5789))
```

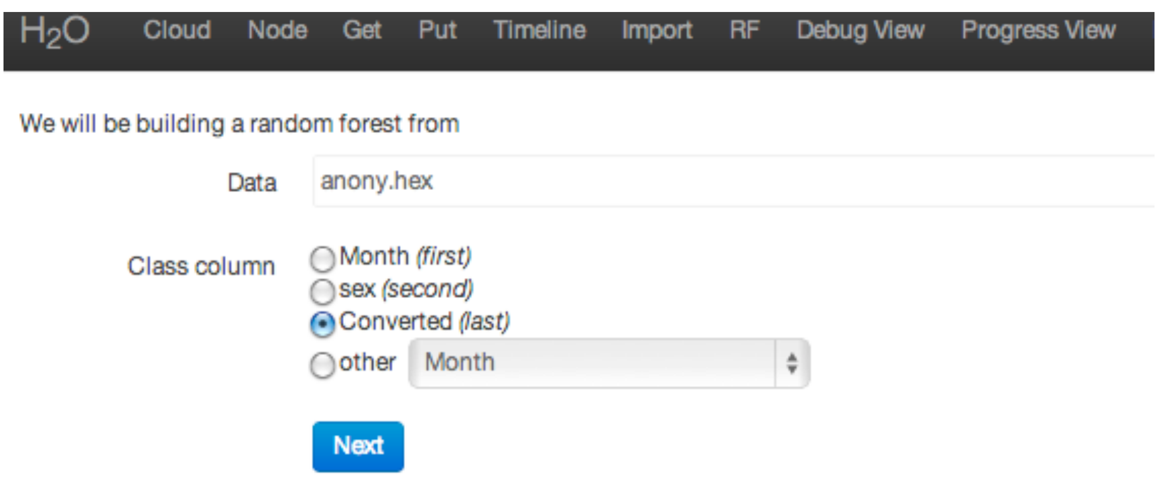
This gives a hint of the features that matter in this particular dataset. The thesis is that features with 0 coefficients, such as "Month", "Sex", "Day of Week" are not useful features in the prediction. Let's see if that thesis holds in the next steps.

Random Forest (RF)

In order to use the Random Forest in H2O Click **RF** Tab – It gives a form for picking dataset.
Using the key name (anony.hex) from the parse page above.



The **RF** query builder prompts you to suggest a class column.



Based on the column the next page allows you to set model parameters Number of Trees, Algorithm (Gini vs. Entropy), Depth, sample Columns to Ignore and Weighting Classes.

We will be building a random forest from

Data	anony.hex
Class column	Converted
Number of trees	5 (default)
Algorithm	Gini
Additional args	(depth, no limit) (bin limit, 1024) (sample, 67) (seed, 181247619891) (features, 9)
Ignore columns	Month sex Day_Week TimeofDay WebApp age AnsweredSurvey Srvy_Plan2DD Srvy_bythngs_online Has_bnk_AC RegisteredOnline Population HouseholdsPerZipCode WhitePopulation BlackPopulation HispanicPopulation AsianPopulation HawaiianPopulation IndianPopulation OtherPopulation MalePopulation FemalePopulation PersonsPerHousehold AverageHouseValue IncomePerHousehold MedianAge MedianAgeMale MedianAgeFemale Elevation CityType division region TimeZone DayLightSaving NumberOfBusinesses NumberOfEmployees BusinessFirstQuarterPayroll BusinessAnnualPayroll GrowthRank GrowthHousingUnits2003 GrowthHousingUnits2004 GrowthIncreaseNumber GrowthIncreasePercentage CBSAPop2003 CBSADivPop2003 DeliveryResidential DeliveryBusiness DeliveryTotal PopulationEstimate LandArea WaterArea id Experian_pass Innovis_pass TU_pass Choicepoint_pass LN_pass Experian_Cx Innovis_Cx TU_Cx Choicepoint_Cx LN_Cx checkpointscore levelonedecisioncode grade white_percent black_percent hispanic_percent male_percent female_percent
Model	model key (default model)
Class weights	(default 1) 0
	(default 1) 1

Calculate Confusion Matrix

Here's the default Random Forest error rate of 0.317

Random Forest of anony.hex
Showing 5 of 5 trees, with 5 trees built
Validate model with another dataset

Model key:**model**

Weighted voting:**default**

Confusion Matrix

Actual \ Predicted	class 0	class 1	Error
class 0	73362	15522	0.175 = 15522 / 88884
class 1	23511	10861	0.684 = 23511 / 34372
Totals	96873	26383	0.317 = 39033 / 123256

Random Decision Trees

min/avg/max depth=131.0 / 165.8 / 207.0, leaves=23551.0 / 24763.8 / 25622.0
Click to view individual trees:

0 1 2 3 4

Increasing the number of trees to 10 improves accuracy.

Random Forest of anony.hex
Showing 10 of 10 trees, with 10 trees built
Validate model with another dataset

Model key:**model**

Weighted voting:**default**

Confusion Matrix

Actual \ Predicted	class 0	class 1	Error
class 0	84005	13814	0.141 = 13814 / 97819
class 1	26914	10900	0.712 = 26914 / 37814
Totals	110919	24714	0.300 = 40728 / 135633

Random Decision Trees

min/avg/max depth=131.0 / 172.2 / 259.0, leaves=23551.0 / 24921.1 / 25947.0
Click to view individual trees:

0 1 2 3 4 5 6 7 8 9

We now "ignore" features that were not significant.
One proposition is that these have strong correlation with features that had 0-coefficients from GLM.

We will be building a random forest from

Data: anony.hex

Class column: Converted

Number of trees: 10

Algorithm: Entropy

Additional args: (depth, no limit) (bin limit, 1024) (sample, 67) (seed, 181

Ignore columns: ☑ Month ☑ sex ☑ Day_Week ☐ TimeofDay ☐ WebApp ☐ age ☐ AnsweredSurvey ☐ Srvy_Plan2DD ☐
Population ☐ HouseholdsPerZipCode ☐ WhitePopulation ☐ BlackPopulation ☐ HispanicPopulati
OtherPopulation ☐ MalePopulation ☐ FemalePopulation ☐ PersonsPerHousehold ☐ AverageHous
MedianAgeFemale ☐ Elevation ☐ CityType ☐ division ☐ region ☐ TimeZone ☐ DayLightSaving ☐ N
BusinessFirstQuarterPayroll ☐ BusinessAnnualPayroll ☐ GrowthRank ☐ GrowthHousingUnits200
GrowthIncreasePercentage ☐ CBSAPop2003 ☐ CBSADivPop2003 ☐ DeliveryResidential ☐ Delive
WaterArea ☐ id ☐ Experian_pass ☐ Innovis_pass ☐ TU_pass ☐ Choicepoint_pass ☐ LN_pass ☐ E
checkpointscore ☐ levelonedecisioncode ☐ grade ☐ white_percent ☐ black_percent ☐ hispanic_pe

Model: model key (default model)

Class weights: (default 1) 0

(default 1) 1

Calculate Confusion Matrix

H2O Cloud Node Get Put Timeline Import RF Debug View Progress View Network Shutdown All

Random Forest of anony.hex
Showing 10 of 10 trees, with 10 trees built
Validate model with another dataset

Model key:**model**

Weighted voting:**default**

Confusion Matrix

Actual \ Predicted	class 0	class 1	Error
class 0	91648	6852	0.070 = 6852 / 98500
class 1	16537	21574	0.434 = 16537 / 38111
Totals	108185	28426	0.171 = 23389 / 136611

Random Decision Trees

min/avg/max depth=106.0 / 157.9 / 235.0, leaves=25816.0 / 26559.8 / 27821.0
Click to view individual trees:

0 1 2 3 4 5 6 7 8 9

An interesting aspect of this dataset is that using entropy as the algorithm gives a model with even better error rate of 0.167 (lower is better)

H2O Cloud Node Get Put Timeline Import RF Debug View Progress View Network Shutdown All

Random Forest of anony.hex
Showing 10 of 10 trees, with 10 trees built
Validate model with another dataset

Model key:**model**

Weighted voting:**default**

Confusion Matrix

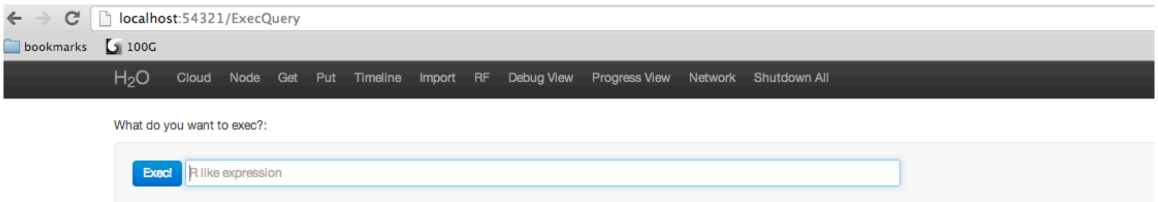
Actual \ Predicted	class 0	class 1	Error
class 0	90401	8099	0.082 = 8099 / 98500
class 1	14656	23455	0.385 = 14656 / 38111
Totals	105057	31554	0.167 = 22755 / 136611

Random Decision Trees

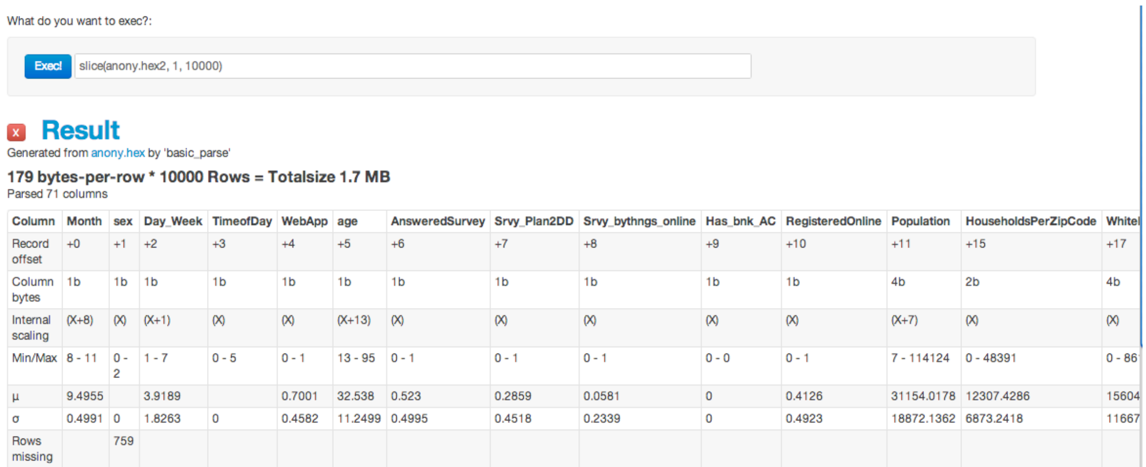
min/avg/max depth=97.0 / 155.3 / 216.0, leaves=24763.0 / 25974.8 / 27028.0
Click to view individual trees:

0 1 2 3 4 5 6 7 8 9

Additional Adhoc Data Manipulation can be achieved via, R or ExecQuery interface:
http://localhost:54321/ExecQuery



Slicing of dataset:
http://localhost:54321/Exec?Expr=slice%28anony.hex2%2C+1%2C+10000%29


What do you want to exec?:

Exec! slice(anony.hex2, 1, 10000)

Result

Generated from anony.hex by 'basic_parse'

179 bytes-per-row * 10000 Rows = Totalsize 1.7 MB

Parsed 71 columns

Column	Month	sex	Day_Week	TimeofDay	WebApp	age	AnsweredSurvey	Srvy_Plan2DD
Record offset	+0	+1	+2	+3	+4	+5	+6	+7
Column bytes	1b	1b	1b	1b	1b	1b	1b	1b
Internal scaling	(X+8)	(X)	(X+1)	(X)	(X)	(X+13)	(X)	(X)
Min/Max	8 - 11	0 - 2	1 - 7	0 - 5	0 - 1	13 - 95	0 - 1	0 - 1
µ	9.4955		3.9189		0.7001	32.538	0.523	0.2859
σ	0.4991	0	1.8263	0	0.4582	11.2499	0.4995	0.4518
Rows missing data		759						

Column	Srvy_bythngs_online	Has_bnk_AC	RegisteredOnline	Population	HouseholdsPerZipCode	Whitel
Record offset	+8	+9	+10	+11	+15	+17
Column bytes	1b	1b	1b	4b	2b	4b
Internal scaling	(X)	(X)	(X)	(X+7)	(X)	(X)
Min/Max	0 - 1	0 - 0	0 - 1	7 - 114124	0 - 48391	0 - 86
µ	0.0581	0	0.4126	31154.0178	12307.4286	15604
σ	0.2339	0	0.4923	18872.1362	6873.2418	11667
Rows missing data						