Project Report

1. Introduction and problem description

Insurance companies are always interested in finding better ways to predict claims severity. The dataset we are going to use is from All State insurance company, which contains 116 categorical variables and 14 continuous variables. People are supposed to predict the severity, which is the loss of a claim, from those 130 independent variables. Kaggle use MAE (Mean Absolute Error) as evaluation metrics.

2. Related work

Insurance companies usually build a parametric probability distribution model from previous claims then predict future claim severity by fitting data into that model.

3. Dataset description

The dataset is from Kaggle competition named "Allstate Claims Severity" (https://www.kaggle.com/c/allstate-claims-severity).

The object is to predict the insurance loss from a dataset from an insurance company.

In the train.csv:
Number of instance: 188319
Number of attributes: 131
Number of category attributes: 116
Number of continuous attributes: 14
Target: loss (continuous variable)

In the test.csv:
Number of instance: 12546
Number of attributes: 130
Number of category attributes: 116
Number of continuous attributes: 14

4. Pre-processing techniques

4.1 Summary the train dataset

```
> summary(train)
       id              cat1       cat2       cat3       cat4       cat5       cat6       cat7       cat8       cat9
 Min.   :     1   A:141550   A:106721   A:177993   A:128395   A:123737   A:131693   A:183744   A:177274   A:113122
 1st Qu.:147748   B: 46768   B: 81597   B: 10325   B: 59923   B: 64581   B: 56625   B:  4574   B: 11044   B: 75196
 Median :294540
 Mean   :294136
 3rd Qu.:440680
 Max.   :587633

 cat10      cat11      cat12      cat13      cat14      cat15      cat16      cat17      cat18      cat19      cat20
 A:160213   A:168186   A:159825   A:168851   A:186041   A:188284   A:181843   A:187009   A:187331   A:186510   A:188114
 B: 28105   B: 20132   B: 28493   B: 19467   B:  2277   B:    34   B:  6475   B:  1309   B:   987   B:  1808   B:   204

 cat21      cat22      cat23      cat24      cat25      cat26      cat27      cat28      cat29      cat30      cat31
 A:187905   A:188275   A:157445   A:181977   A:169969   A:177119   A:168250   A:180938   A:184593   A:184760   A:182980
 B:   413   B:    43   B: 30873   B:  6341   B: 18349   B: 11199   B: 20068   B:  7380   B:  3725   B:  3558   B:  5338

 cat32      cat33      cat34      cat35      cat36      cat37      cat38      cat39      cat40      cat41      cat42
 A:187107   A:187361   A:187734   A:188105   A:156313   A:165729   A:169323   A:183393   A:180119   A:181177   A:186623
 B:  1211   B:   957   B:   584   B:   213   B: 32005   B: 22589   B: 18995   B:  4925   B:  8199   B:  7141   B:  1695

cat43      cat44      cat45      cat46      cat47      cat48      cat49      cat50      cat51      cat52      cat53
A:184110   A:172716   A:183991   A:187436   A:187617   A:188049   A:179127   A:137611   A:187071   A:179505   A:172949
B:  4208   B: 15602   B:  4327   B:   882   B:   701   B:   269   B:  9191   B: 50707   B:  1247   B:  8813   B: 15369

cat54      cat55      cat56      cat57      cat58      cat59      cat60      cat61      cat62      cat63      cat64
A:183762   A:188173   A:188136   A:185296   A:188079   A:188018   A:187872   A:187596   A:188273   A:188239   A:188271
B:  4556   B:   145   B:   182   B:  3022   B:   239   B:   300   B:   446   B:   722   B:    45   B:    79   B:    47

cat65      cat66      cat67      cat68      cat69      cat70      cat71      cat72      cat73      cat74      cat75
A:186056   A:179982   A:187626   A:188176   A:188011   A:188295   A:178646   A:118322   A:154275   A:184731   A:154307
B:  2262   B:  8336   B:   692   B:   142   B:   307   B:    23   B:  9672   B: 69996   B: 34017   B:  3561   B: 34010
                                                                                        C:    26   C:    26   C:     1

cat76      cat77      cat78      cat79      cat80      cat81      cat82      cat83      cat84      cat85      cat86
A:181347   A:    49   A:   788   A:  7064   A:   783   A:   788   A: 19322   A: 26038   A: 29450   A:   788   A:  1589
B:  6183   B:   358   B:186526   B:152929   B: 46538   B: 24132   B:147536   B:141534   B:   431   B:186005   B:103852
C:   788   C:   408   C:   645   C:  1668   C:  3492   C:  9013   C:  2655   C:  4958   C:154939   C:  1011   C: 10290
           D:187503   D:   359   D: 26657   D:137505   D:154385   D: 18805   D: 15788   D:  3498   D:   514   D: 72587
```

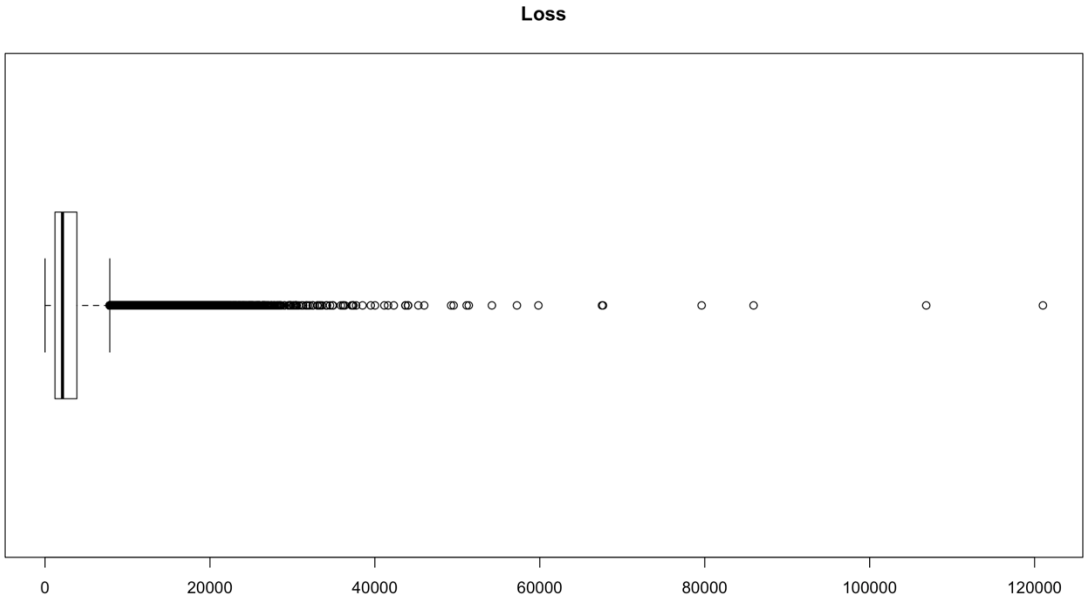
```
cat87      cat88          cat89        cat90          cat91        cat92      cat93      cat94      cat95
A:   788   A:168926   A      :183744   A:177993   A      :111028   A:124689   A:   432   A:   738   A: 3736
B:166992   B:     7   B      :  4312   B:  9515   B      : 42630   B:   628   B:  1133   B: 51710   B:  109
C:  8819   D: 19302   C      :   220   C:   728   G      : 26734   C:    62   C: 35788   C: 13623   C:87531
D: 11719   E:    83   D      :    33   D:    70   C      :  6400   D:    11   D:150237   D:121642   D:79525
                      E      :     5   E:     6   D      :  1149   F:     1   E:   728   E:    91   E:17417
                      I      :     2   F:     4   E      :   254   H: 62901              F:   494
                      (Other):     2   G:     2   (Other):   123   I:    26              G:    20
    cat96        cat97     cat98          cat99          cat100          cat101          cat102          cat103
E      :174360   A:41970   A:105492   P      :79455   F      :42970   A      :106721   A      :177274   A      :123737
D      :  7922   B:   34   B:   542   T      :72591   I      :39933   D      : 17171   B      :  5155   B      : 33342
B      :  2957   C:78127   C: 21485   R      :10290   L      :19961   C      : 16971   C      :  4929   C      : 16508
G      :  2665   D: 3779   D: 50557   D      : 8844   K      :13817   G      : 10944   E      :   482   D      :  7806
F      :   343   E:47450   E: 10242   S      : 7045   G      :12935   F      : 10139   D      :   449   E      :  4473
A      :    35   F:  213              N      : 2894   J      :12027   J      :  7259   G      :    15   F      :  1528
(Other):    36   G:16745              (Other): 7199   (Other):46675   (Other): 19113   (Other):    14   (Other):   924
    cat104          cat105          cat106          cat107          cat108          cat109          cat110
E      :42925   E      :76493   G      :47165   F      :47310   B      :65512   BI     :152918   CL     :25305
G      :40660   F      :62892   H      :37713   G      :28560   K      :42435   AB     : 21933   EG     :24654
D      :27611   G      :20613   F      :36143   H      :23461   G      :21421   BU     :  3142   CS     :24592
F      :19228   D      :12172   I      :21433   J      :22405   D      :19160   K      :  2999   EB     :21396
H      :17187   H      :11258   J      :18281   K      :20236   F      :10242   G      :  1353   CO     :17495
K      :14297   I      : 2941   E      :13000   I      :20066   A      : 9299   BQ     :  1067   BT     :16365
(Other):26410   (Other): 1949   (Other):14583   (Other):26280   (Other):20249   (Other):  4906   (Other):58511
    cat111          cat112          cat113          cat114          cat115          cat116          cont1
A      :128395   E      :25148   BM     :26191   A      :131693   K      :43866   HK     : 21061   Min.   :0.000016
C      : 32401   AH     :18639   AE     :22030   C      : 16793   O      :26813   DJ     : 20244   1st Qu.:0.346090
E      : 14682   AS     :17669   L      :13058   E      : 16475   J      :23895   CK     : 10162   Median :0.475784
G      :  7039   J      :16222   AX     :12661   J      :  8199   N      :22438   DP     :  9202   Mean   :0.493861
I      :  3578   AF     : 9368   Y      :11374   F      :  7905   P      :21538   GS     :  8736   3rd Qu.:0.623912
K      :  1353   AN     : 9138   K      : 7738   N      :  2455   L      :16125   CR     :  6862   Max.   :0.984975
(Other):   870   (Other):92134   (Other):95266   (Other):  4798   (Other):33643   (Other):112051

    cont2             cont3             cont4             cont5             cont6             cont7             cont8
Min.   :0.001149   Min.   :0.002634   Min.   :0.1769   Min.   :0.2811   Min.   :0.01268   Min.   :0.0695   Min.   :0.2369
1st Qu.:0.358319   1st Qu.:0.336963   1st Qu.:0.3274   1st Qu.:0.2811   1st Qu.:0.33610   1st Qu.:0.3502   1st Qu.:0.3128
Median :0.555782   Median :0.527991   Median :0.4529   Median :0.4223   Median :0.44094   Median :0.4383   Median :0.4411
Mean   :0.507188   Mean   :0.498918   Mean   :0.4918   Mean   :0.4874   Mean   :0.49094   Mean   :0.4850   Mean   :0.4864
3rd Qu.:0.681761   3rd Qu.:0.634224   3rd Qu.:0.6521   3rd Qu.:0.6433   3rd Qu.:0.65502   3rd Qu.:0.5910   3rd Qu.:0.6236
Max.   :0.862654   Max.   :0.944251   Max.   :0.9543   Max.   :0.9837   Max.   :0.99716   Max.   :1.0000   Max.   :0.9802

    cont9             cont10            cont11            cont12            cont13            cont14
Min.   :0.00008   Min.   :0.0000   Min.   :0.03532   Min.   :0.03623   Min.   :0.000228   Min.   :0.1797
1st Qu.:0.35897   1st Qu.:0.3646   1st Qu.:0.31096   1st Qu.:0.31166   1st Qu.:0.315758   1st Qu.:0.2946
Median :0.44145   Median :0.4612   Median :0.45720   Median :0.46229   Median :0.363547   Median :0.4074
Mean   :0.48551   Mean   :0.4981   Mean   :0.49351   Mean   :0.49315   Mean   :0.493138   Mean   :0.4957
3rd Qu.:0.56682   3rd Qu.:0.6146   3rd Qu.:0.67892   3rd Qu.:0.67576   3rd Qu.:0.689974   3rd Qu.:0.7246
Max.   :0.99540   Max.   :0.9950   Max.   :0.99874   Max.   :0.99848   Max.   :0.988494   Max.   :0.8448

     loss
Min.   :     0.67
1st Qu.:  1204.46
Median :  2115.57
Mean   :  3037.34
3rd Qu.:  3864.05
Max.   :121012.25
```

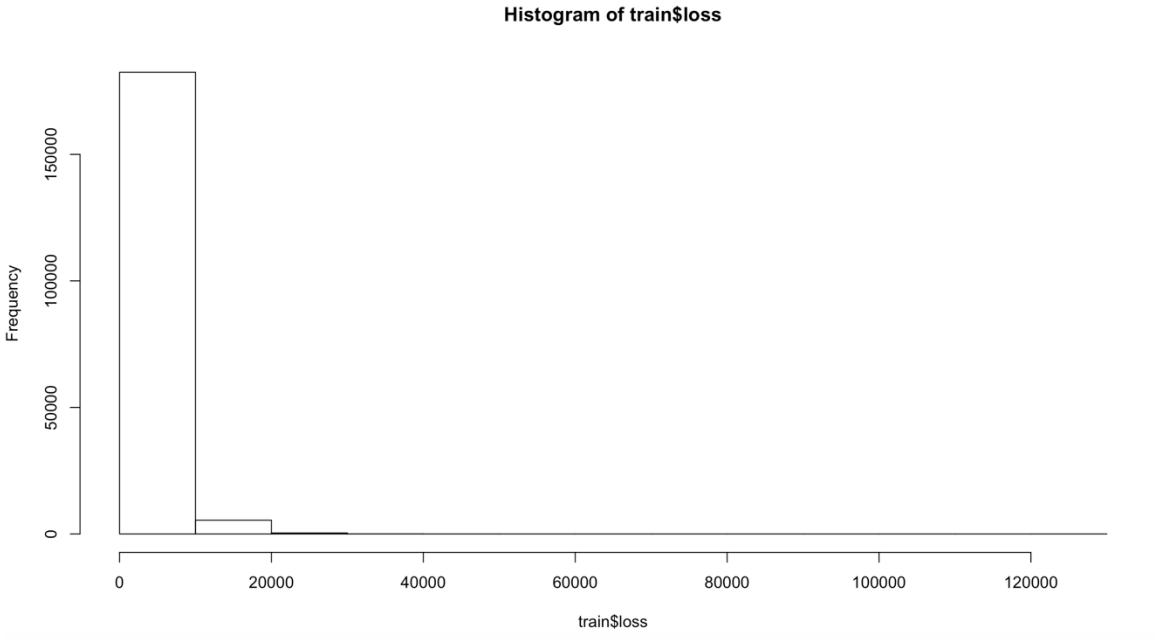
We can see:

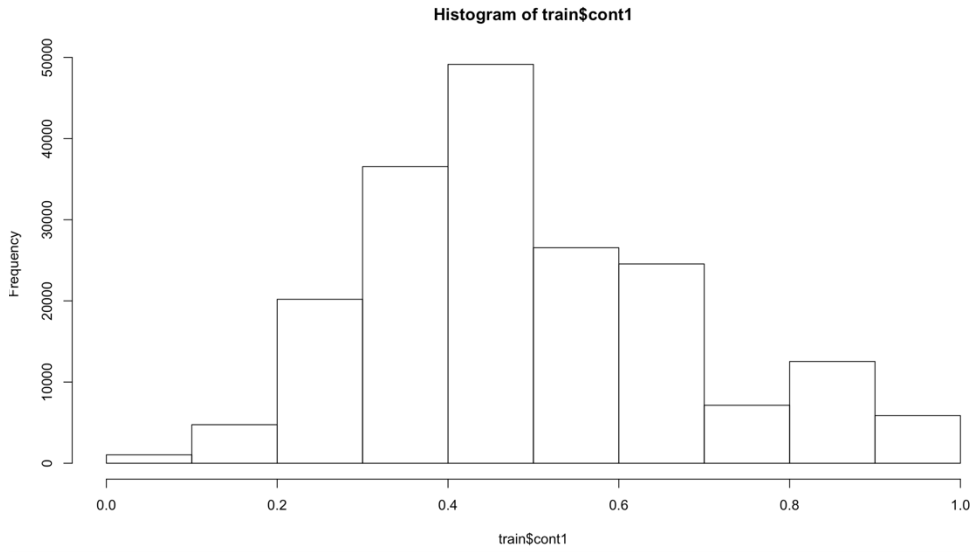
All attributes are present and all rows can be used.

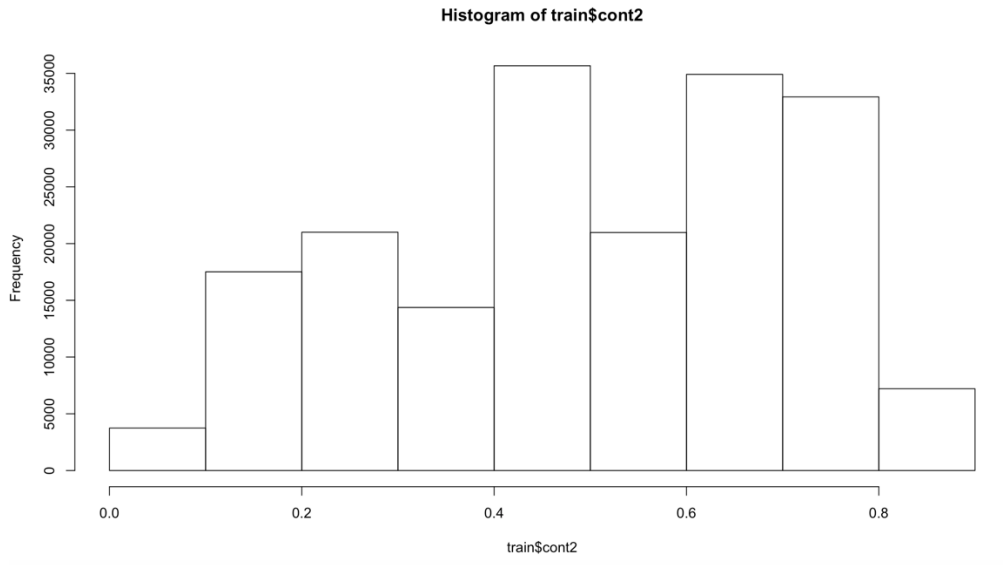
Neither null value nor “?” are present.

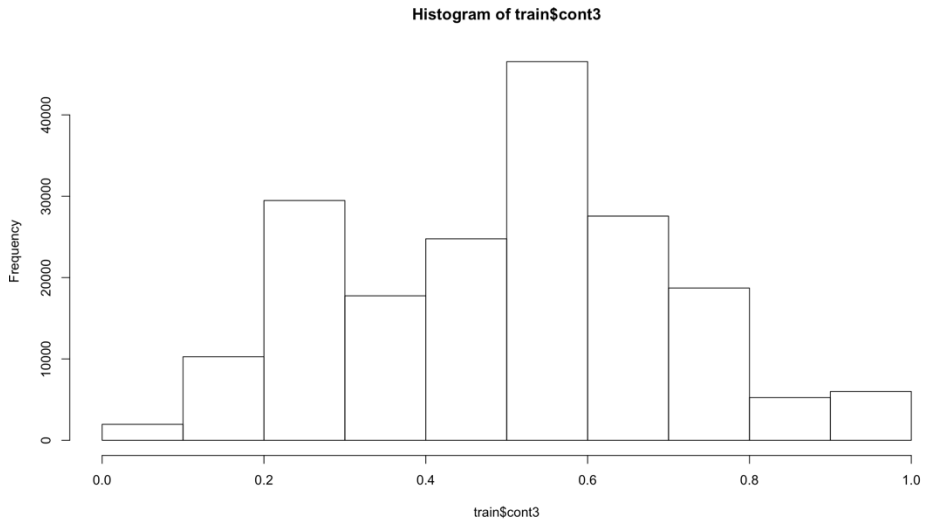
4.2 Data visualization for continuous attributes

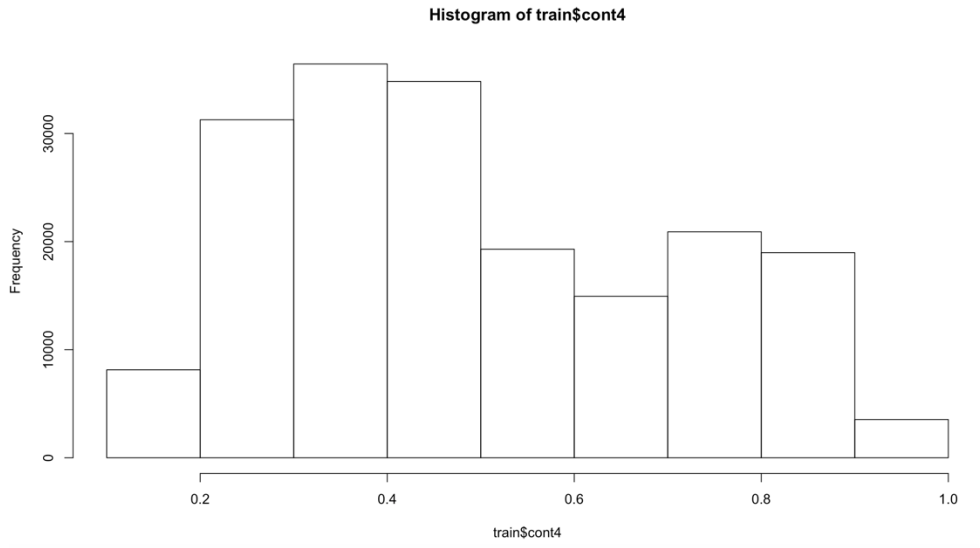


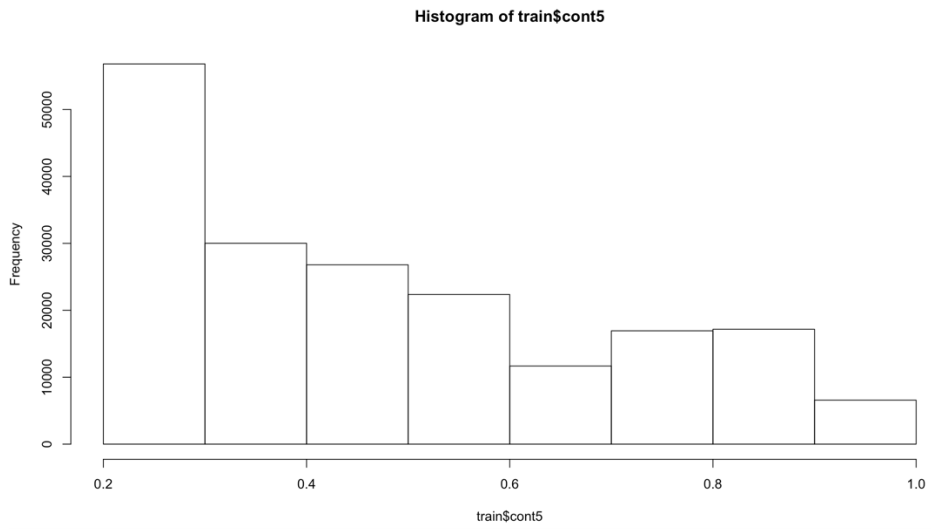


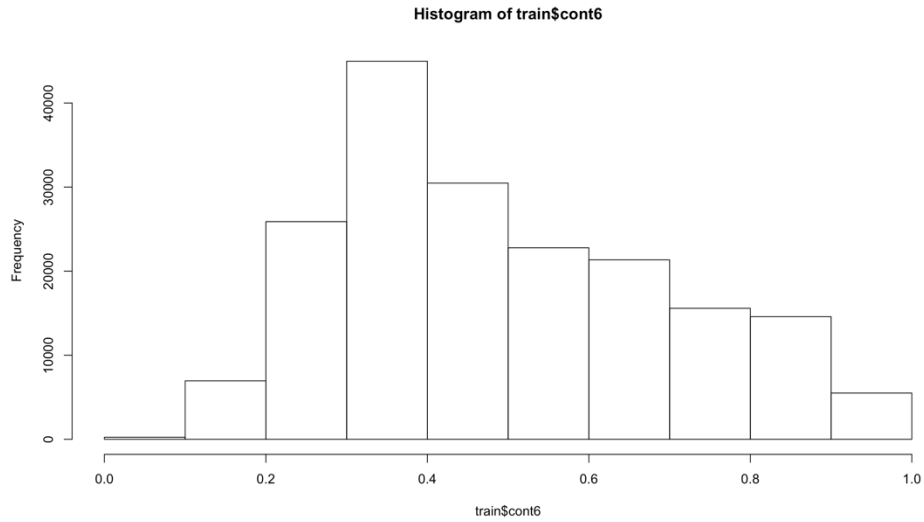

Histogram of train$cont1
Frequency
0
10000
20000
30000
40000
50000
0.0
0.2
0.4
0.6
0.8
1.0
train$cont1

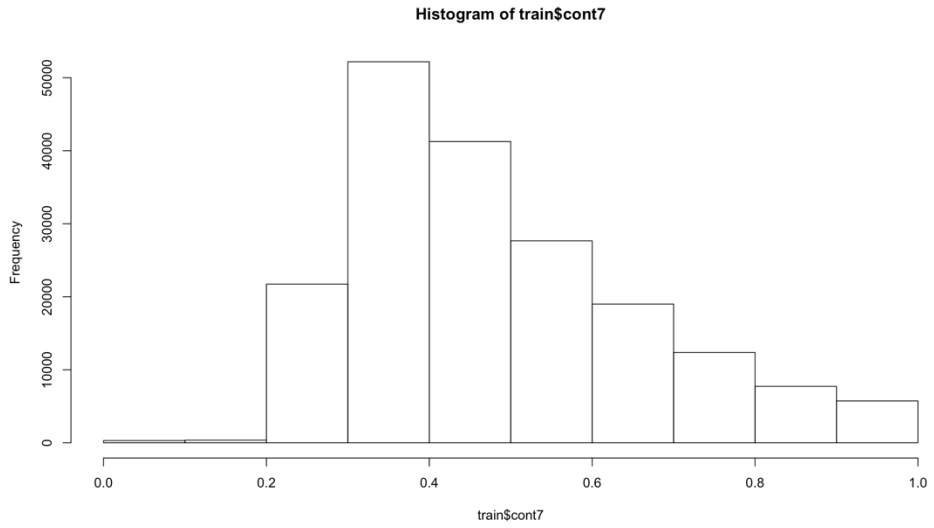

Histogram of train$cont2
Frequency
0
5000
10000
15000
20000
25000
30000
35000
0.0
0.2
0.4
0.6
0.8
train$cont2

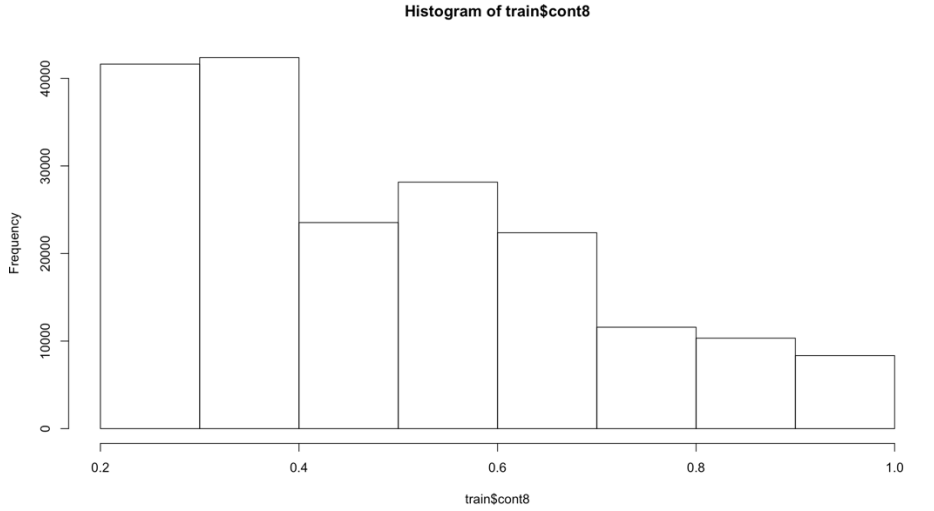

Histogram of train$cont3
Frequency
0
10000
20000
30000
40000
0.0
0.2
0.4
0.6
0.8
1.0
train$cont3

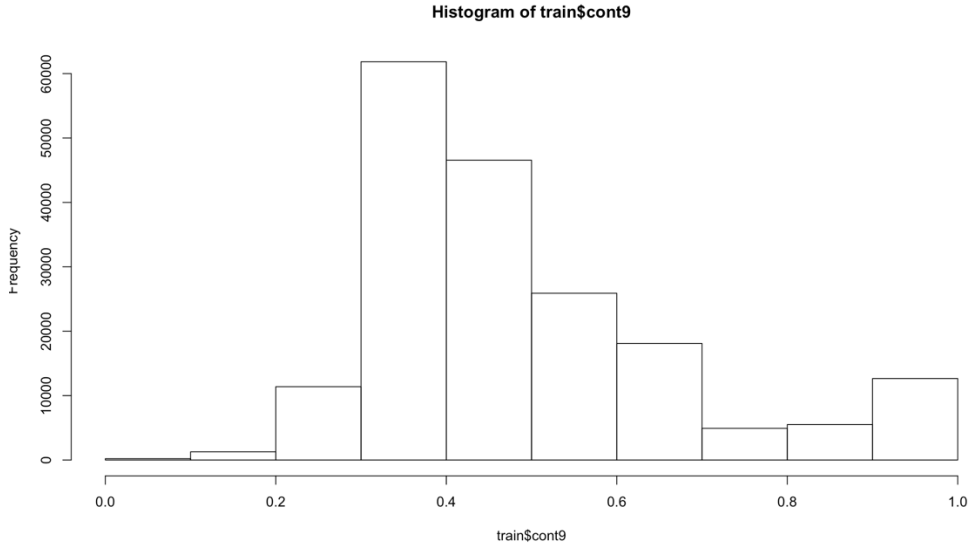

Histogram of train$cont4
Frequency
0
10000
20000
30000
0.2
0.4
0.6
0.8
1.0
train$cont4

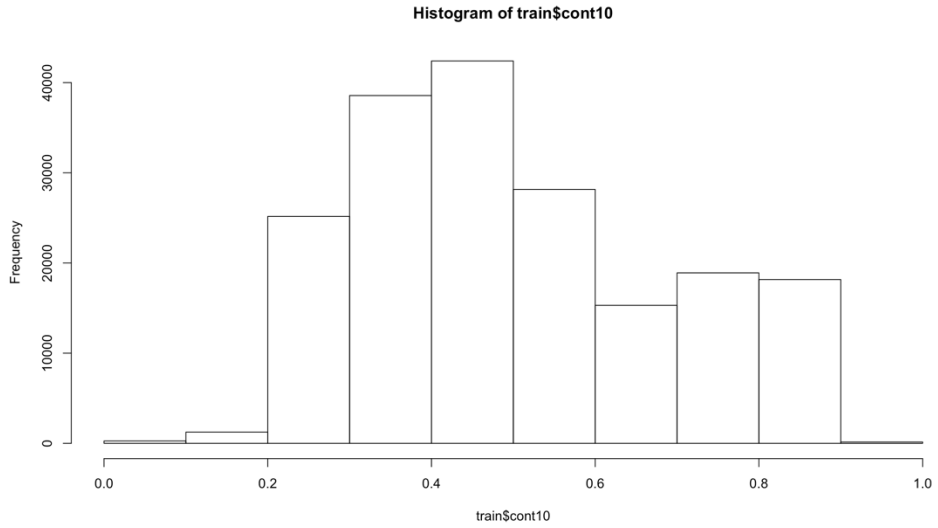

Histogram of train$cont5
Frequency
0
10000
20000
30000
40000
50000
0.2
0.4
0.6
0.8
1.0
train$cont5

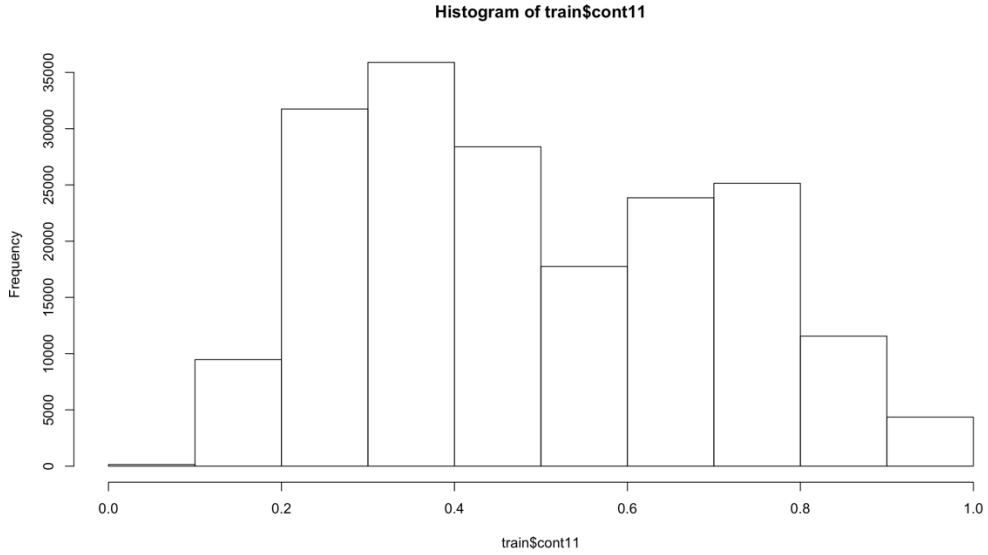

Histogram of train$cont6
Frequency
0
10000
20000
30000
40000
0.0
0.2
0.4
0.6
0.8
1.0
train$cont6

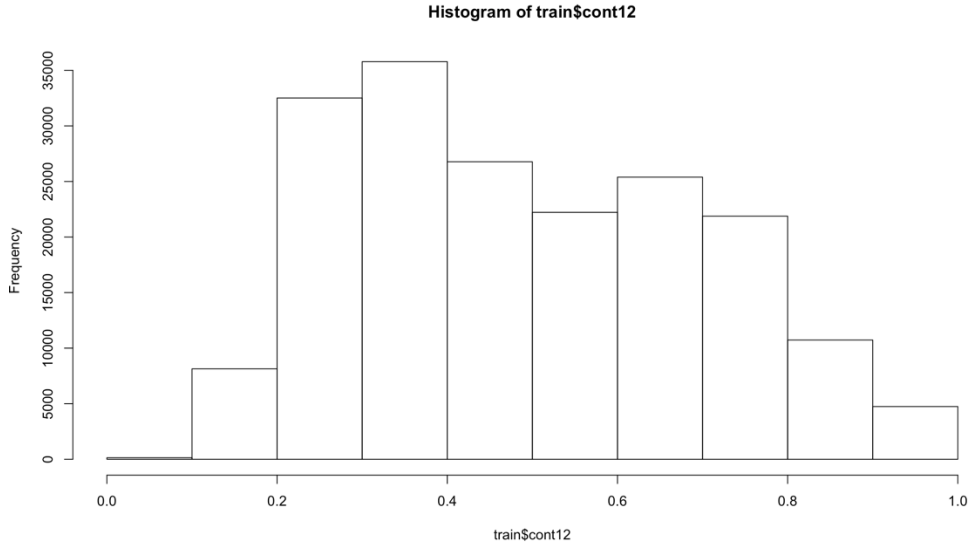

Histogram of train$cont7
Frequency
0
10000
20000
30000
40000
50000
0.0
0.2
0.4
0.6
0.8
1.0
train$cont7

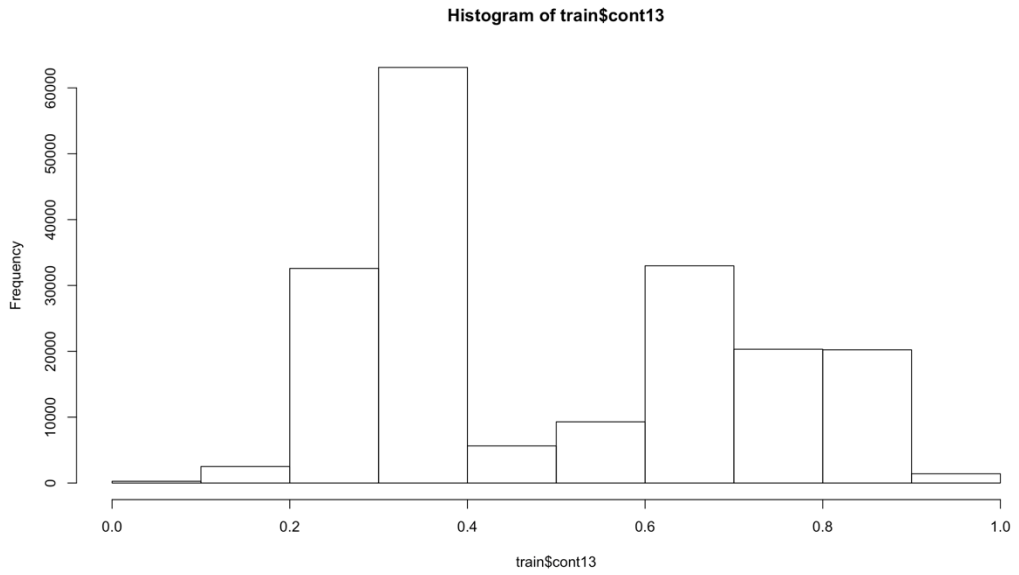

Histogram of train$cont8
Frequency
0
10000
20000
30000
40000
0.2
0.4
0.6
0.8
1.0
train$cont8

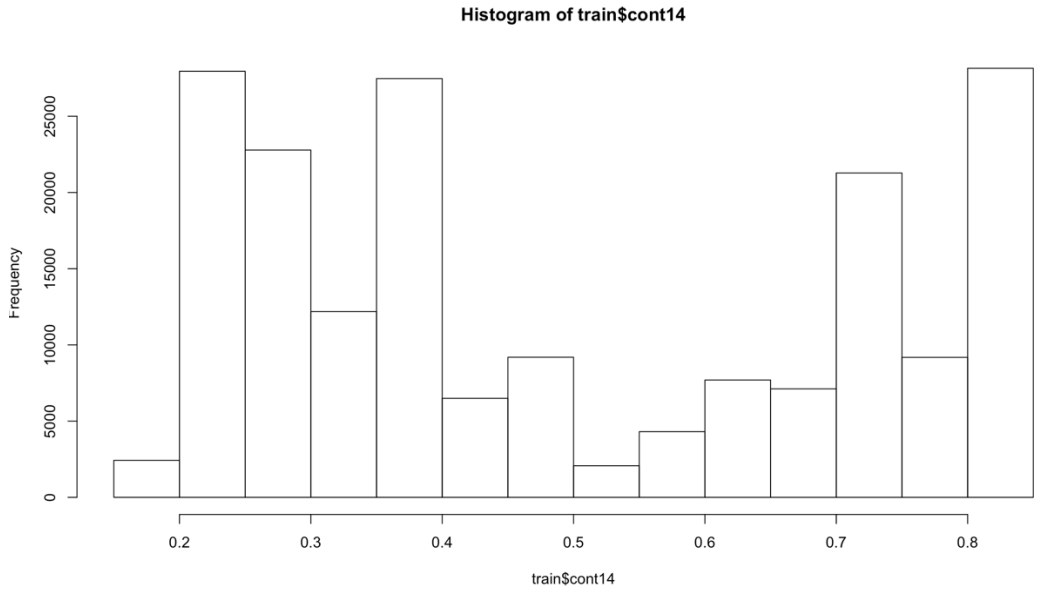

Histogram of train$cont9
Frequency
0
10000
20000
30000
40000
50000
60000
0.0
0.2
0.4
0.6
0.8
1.0
train$cont9


Histogram of train$cont10
Frequency
0
10000
20000
30000
40000
0.0
0.2
0.4
0.6
0.8
1.0
train$cont10


Histogram of train$cont11
Frequency
0
5000
10000
15000
20000
25000
30000
35000
0.0
0.2
0.4
0.6
0.8
1.0
train$cont11


Histogram of train$cont12
Frequency
0
5000
10000
15000
20000
25000
30000
35000
0.0
0.2
0.4
0.6
0.8
1.0
train$cont12





We can see that:

Most of cont1's value close to 0.5 and most of cont5's value close to 0.3.

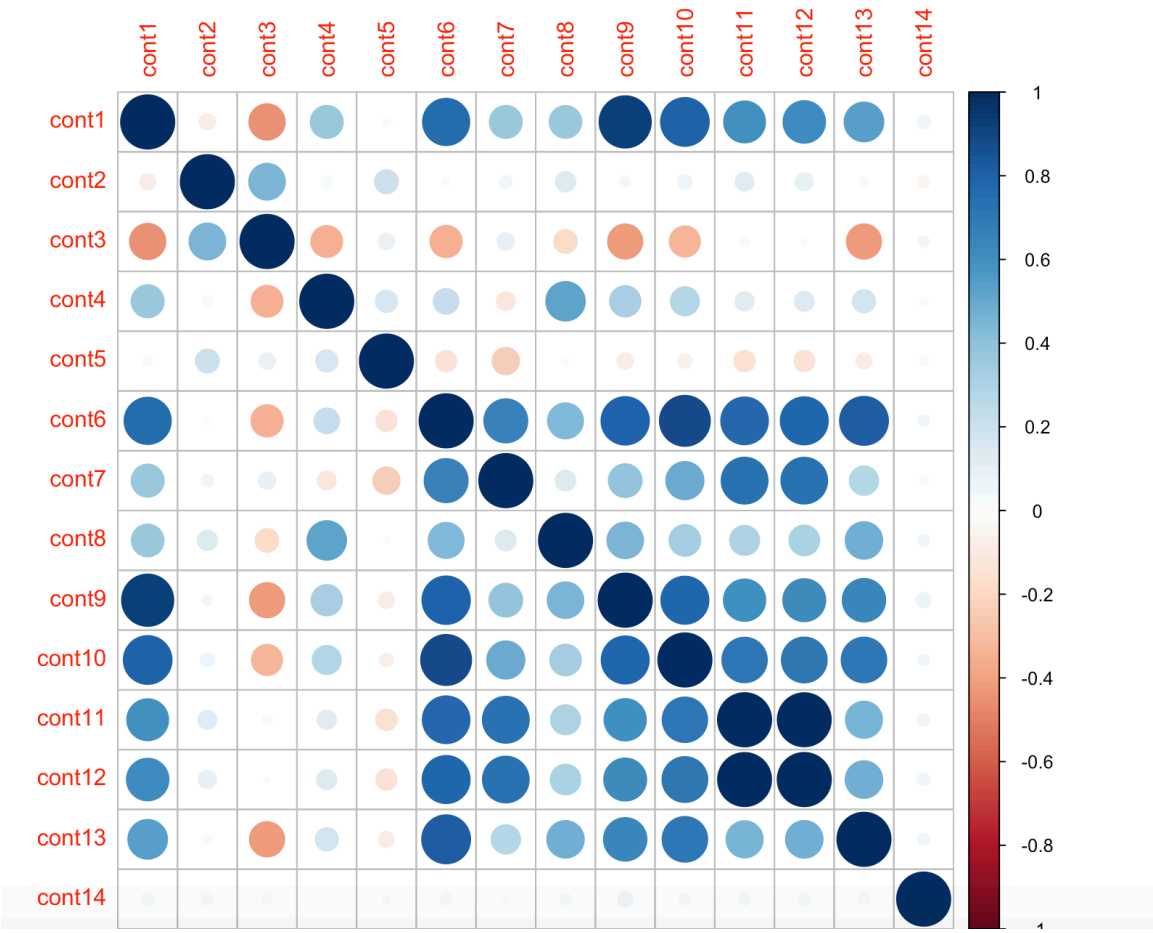
Cont14 and cont2 look like spikes at specific points.

The 'loss' distribution is strange, not a normal distribution. Most of the 'loss' value is about 10000.

4.3 Data correlation and scatter plot of continuous attributes

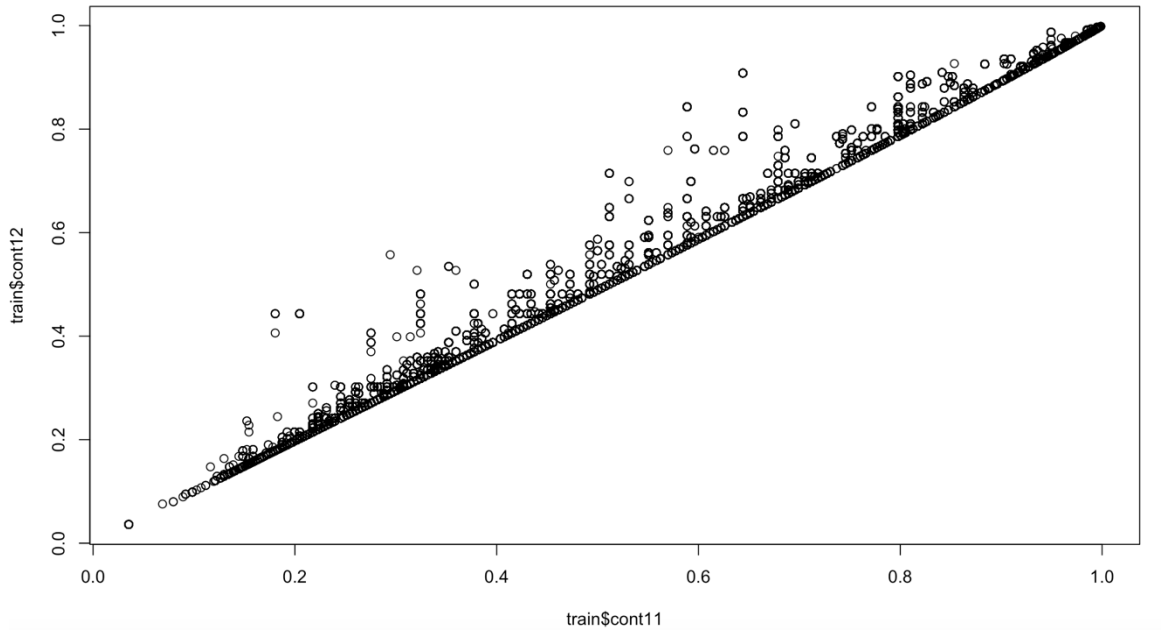
We use correlation matrix and plot to find high correlated attributes:

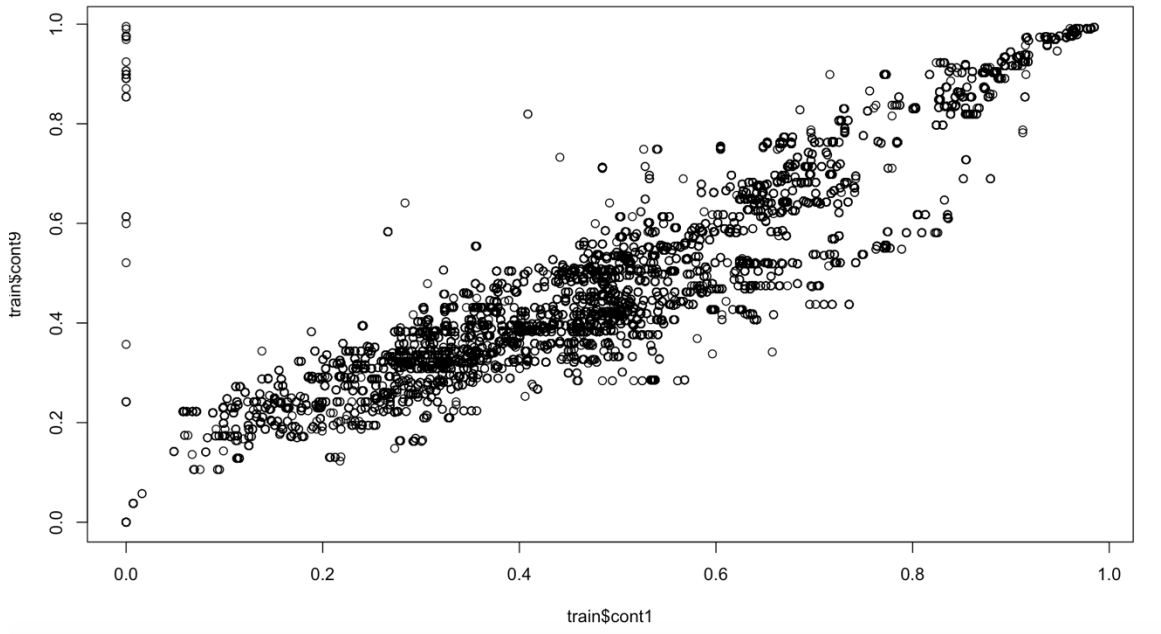
```
> corMatrix
              cont1       cont2        cont3       cont4       cont5       cont6       cont7
cont1   1.00000000 -0.08518029 -0.445431486  0.36754922 -0.02522996  0.75831532  0.36738447
cont2  -0.08518029  1.00000000  0.455860923  0.03869311  0.19142746  0.01586389  0.04818716
cont3  -0.44543149  0.45586092  1.000000000 -0.34163320  0.08941736 -0.34927774  0.09751599
cont4   0.36754922  0.03869311 -0.341633205  1.00000000  0.16374769  0.22093229 -0.11506357
cont5  -0.02522996  0.19142746  0.089417360  0.16374769  1.00000000 -0.14981040 -0.24934373
cont6   0.75831532  0.01586389 -0.349277744  0.22093229 -0.14981040  1.00000000  0.65891830
cont7   0.36738447  0.04818716  0.097515992 -0.11506357 -0.24934373  0.65891830  1.00000000
cont8   0.36116252  0.13746777 -0.185432316  0.52874030  0.00901470  0.43743713  0.14204208
cont9   0.92991171 -0.03272913 -0.417054053  0.32896064 -0.08820196  0.79754352  0.38434291
cont10  0.80855087  0.06352640 -0.325562044  0.28329422 -0.06496684  0.88335051  0.49262071
cont11  0.59608980  0.11682355  0.025270913  0.12092661 -0.15154819  0.77374545  0.74710792
cont12  0.61422545  0.10625016  0.006110642  0.13045303 -0.14821668  0.78514397  0.74271226
cont13  0.53484952  0.02333465 -0.418202580  0.17934193 -0.08291452  0.81509108  0.28839491
cont14  0.05668837 -0.04558425 -0.039592319  0.01744543 -0.02163780  0.04217772  0.02228598
              cont8       cont9       cont10      cont11       cont12      cont13      cont14
cont1   0.36116252  0.92991171  0.80855087  0.59608980  0.614225455  0.53484952  0.05668837
cont2   0.13746777 -0.03272913  0.06352640  0.11682355  0.106250164  0.02333465 -0.04558425
cont3  -0.18543232 -0.41705405 -0.32556204  0.02527091  0.006110642 -0.41820258 -0.03959232
cont4   0.52874030  0.32896064  0.28329422  0.12092661  0.130453027  0.17934193  0.01744543
cont5   0.00901470 -0.08820196 -0.06496684 -0.15154819 -0.148216682 -0.08291452 -0.02163780
cont6   0.43743713  0.79754352  0.88335051  0.77374545  0.785143972  0.81509108  0.04217772
cont7   0.14204208  0.38434291  0.49262071  0.74710792  0.742712263  0.28839491  0.02228598
cont8   1.00000000  0.45265753  0.33658772  0.30238054  0.315904152  0.47640159  0.04353935
cont9   0.45265753  1.00000000  0.78569679  0.60800047  0.626656437  0.64202769  0.07415381
cont10  0.33658772  0.78569679  1.00000000  0.70289554  0.713811933  0.70787639  0.04180832
cont11  0.30238054  0.60800047  0.70289554  1.00000000  0.994384110  0.46624667  0.04729287
cont12  0.31590415  0.62665644  0.71381193  0.99438411  1.000000000  0.47867667  0.05026658
cont13  0.47640159  0.64202769  0.70787639  0.46624667  0.478676675  1.00000000  0.04754275
cont14  0.04353935  0.07415381  0.04180832  0.04729287  0.050266580  0.04754275  1.00000000
```



The high correlation between cont11 and cont12 is 0.99.

The high correlation between cont1 and cont9 is 0.93.



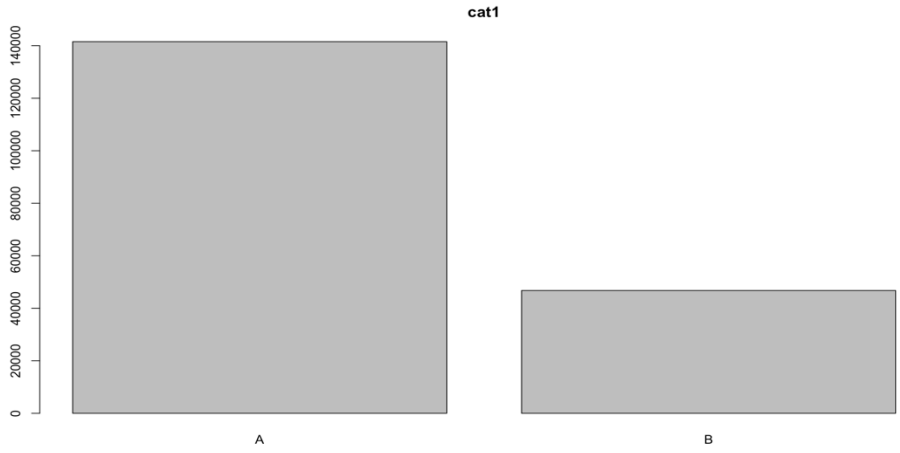


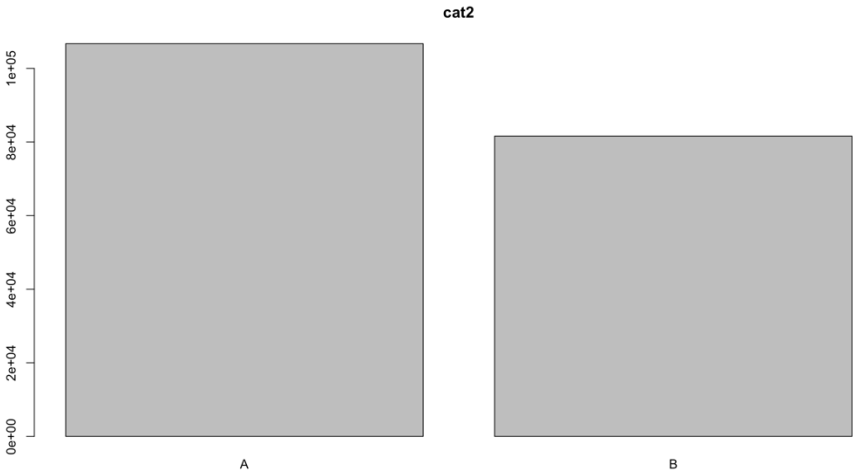
We can see that:

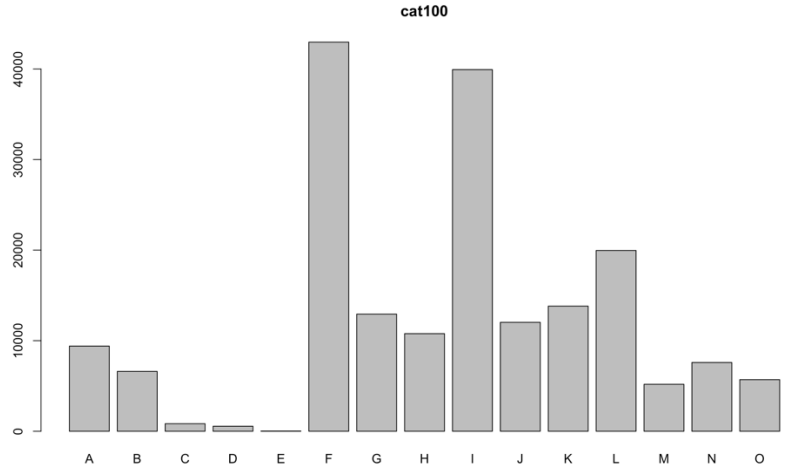
Cont11 and cont12 are highly correlated. There is a linear model between cont11 and cont12. It is not good to keep the both attributes. One of the two attributes must be deleted.

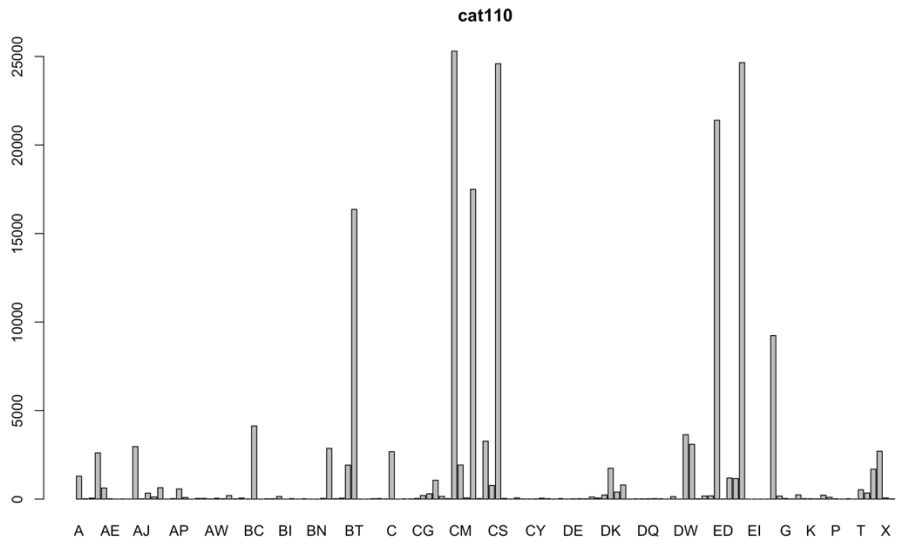
Cont1 and cont9 are also highly correlated. There is a linear model between cont1 and cont9. It is also not good to keep the both attributes. One of the two attributes must be deleted.

4.4 Data visualization for categorical attributes









…… (116 plots were built for categorical attributes visualization)

We can see that:

From cat1 to cat72, there are only 2 labels in each attribute (A and B).

From cat73 to cat76, there are 3 labels in each attribute.

From cat77 to cat88, there are 4 labels in each attribute.

From cat89 to cat116, there are many labels in each attribute.

5. Proposed solution and methods

Since the target variable ('loss') is continuous, regression methods such as linear regression, LASSO (Least Absolute Shrinkage and Selection Operator) or GBR (Gradient Boosting Regressor) will be applied to this dataset.

5.1 Linear Regression

We all know linear regression well. We could build a linear regression model with all attributes. Then the model would be evaluated by MAE and MSE. We would also build a linear regression model with various combination of attributes. By deleting useless attributes, we predict this model would have better evaluation result.

5.2 Ridge Regression

To prevent overfitting, we would try to build a ridge regression model.

The alpha value is very important for the ridge regression. We need to try several different alpha value to find a model with smallest MAE. We also would build the ridge regression model with various combination of attributes.

5.3 LASSO Linear Regression

Ridge regression improves prediction error and reduce overfitting, but it does not do covariate selection and therefore cannot make the model more interpretable. In order to improve this, we would build a LASSO linear regression model. The main step is much the same as ridge regression.

5.4 AdaBoost

AdaBoost is an algorithm for constructing a “strong” classifier as linear combination of simple “weak” classifiers.

5.5 Random Forest

Random Forest is a refinement of bagged trees. When the tree split, a random sample of some features is drawn and only those chosen features are considered for tree splitting.

5.6 Bagging

Bootstrap aggregating (Bagging) is an algorithm designed to improve the stability and accuracy of classification and regression by reducing variance and helping to avoid overfitting.

5.7 Gradient Boosting

The n estimator parameter is added to build a gradient boosting model. We would check different n value to build a better model with lower MAE.

5.8 Extreme Gradient Boosting

Extreme Gradient Boosting (xgboost) is a modified version of gradient boosting.

6. Evaluation techniques

As the kaggle required, mean absolute error (MAE) is only used to evaluate the accuracy of model:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| = \frac{1}{n}\sum_{i=1}^{n}\left|e_i\right| .$$

Where

$$AE = |e_i| = |y_i - \hat{y}_i|$$

$$Actual = y_i$$

$$Predicted = \hat{y}_i$$

7. Coding language / technique to be used

R will be used for data visualization.

Python will be used for the rest of all works such as data pre-cleaning, processing, model building and predicting.

8. Experimental results and analysis

Since the distribution of target variable "loss" is not normal, so the first step is transform that variable. We used log transformation and found shift value 1500 works best by parameter tuning.

Loss = log_{10} (loss + shift)

Shift	MAE
50	1280.57149767
100	1277.98974664
150	1275.82563592
200	1273.97832532
250	1272.38697447
300	1271.00173961
1000	1262.74242893
1500	**1262.39066045**
1750	1262.80001391
2000	1263.38130528
5000	1275.38232042

Then we applied each model to the training dataset.

In order to find the best attributes for the linear model, we did the stepwise regression in R. But because of the huge dataset, the stepwise regression models took more than 24 hours to build and not be successful. We decide to use the full attributes model.

7 models were built to predict 'loss' by Python:

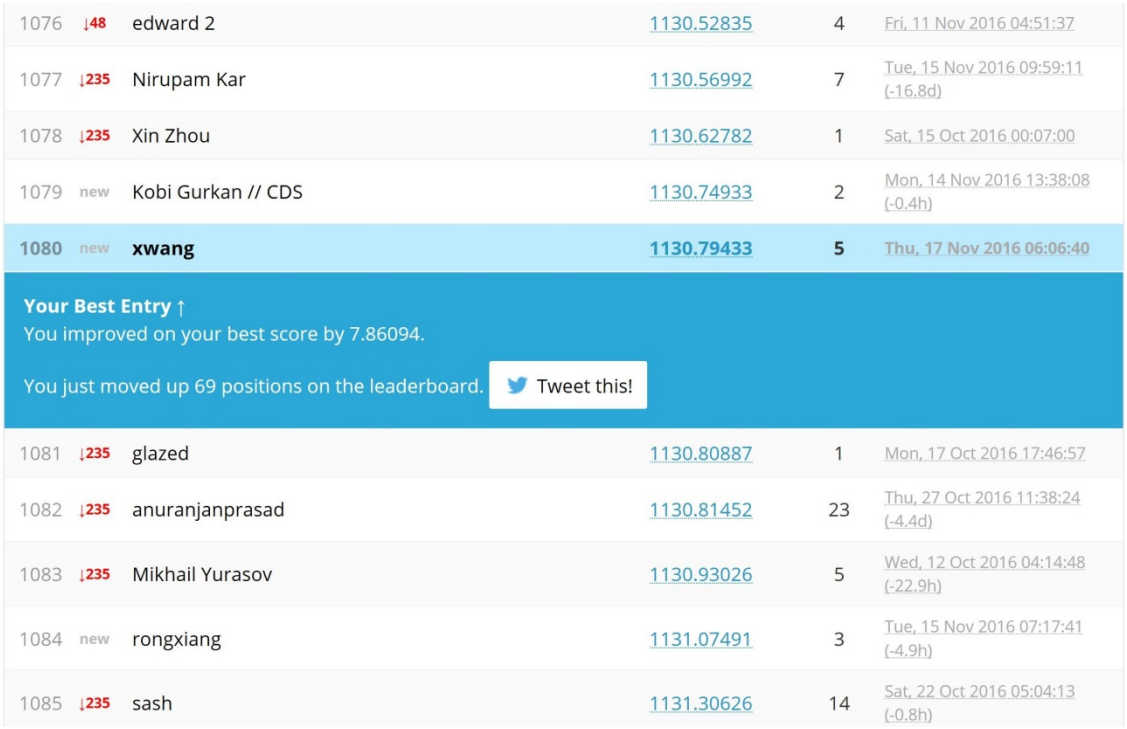
Classifier	Para1	Para2	Para3	MAE
Linear Regression				1262.39
Ridge Regression				1262.30
LASSO Regression				1257.91
AdaBoost	n_estimators=50	learning_rate=1.0	loss='linear'	1705.48
AdaBoost	n_estimators=50	learning_rate=0.1	loss='linear'	1463.55
AdaBoost	n_estimators=100	learning_rate=0.1	loss='linear'	1491.76
AdaBoost	n_estimators=50	learning_rate=0.1	loss='square'	1489.18
AdaBoost	n_estimators=10	learning_rate=0.1	loss='linear'	1454.44
Random Forest	n_estimators=10	max_depth=None	criterion='mse'	1276.87
Random Forest	n_estimators=100	max_depth=None	criterion='mse'	1221.81
Random Forest	n_estimators=10	max_depth=None	criterion='mae'	NA
Random Forest	n_estimators=10	max_depth=5	criterion='mse'	1381.23
Bagging	n_estimators=10			1276.92
Bagging	n_estimators=100			1222.12
Gradient Boosting	n_estimators=10			1509.35
Gradient Boosting	n_estimators=100			1215.56

Xgboost

booster	eta	gamma	min_child_weight	max_depth	col_bytree	round	MAE
gbtree	1	1	1	1	1	1000	1237.06
gblinear	1	1	1	1	1	1000	1262.18
dart	1	1	1	1	1	1000	1237.06
gbtree	0.5	1	1	1	1	1000	1235.43
gbtree	0.1	1	1	1	1	1000	1239.43
gbtree	0.2	1	1	1	1	1000	1238.13
gbtree	0.01	1	1	1	1	1000	1363.33
gbtree	0.1	1	1	1	1	10000	1239.43
gbtree	0.1	0.5	1	1	1	10000	1231.97
gbtree	0.1	0.25	1	1	1	10000	1226.87
gbtree	0.1	0.125	1	1	1	10000	1223.45

gbtree	0.1	0.01	1	1	1	10000	1220.75
gbtree	0.1	0	1	1	1	10000	1220.16
gbtree	0.1	0	1	6	1	10000	1255.37
gbtree	0.1	0	1	5	1	10000	1209.78
gbtree	0.1	0	1	4	1	10000	1225.56
gbtree	0.1	0	1	5	0.5	10000	1204.71
gbtree	0.1	0	1	5	0.25	5000	1179.26
gbtree	0.1	0	1	5	0.25	2500	1167.47
gbtree	0.1	0	1	5	0.25	2000	1164.08
gbtree	0.1	0	1	5	0.25	1000	1158.73
gbtree	0.1	0	1	5	0.5	1000	1159.56
gbtree	0.1	0	2	5	0.1	1000	1160.69

Kaggle submission rank:



9. Conclusion

9.1 Parameter tuning is very important for model building.

9.2 Xgboost is extremely powerful for supervised learning for which it has dominated the Kaggle competition.

9.3 We wish to use stepwise regression to find best attributes for linear model, but due to the big dataset, we couldn't do it successfully on R. We decide to use full attributes.

9.4 Attributes cont12 and cont9 were deleted because of high correlation and new MAE was calculated:

Classifier	Para1	Para2	Para3	MAE	MAE(delete)	
Linear Regression				1262.39	1262.90	↑ 0.511
Ridge Regression				1262.30	1262.765	↑ 0.461
LASSO Regression				1257.916	1257.904	↓0.0123
AdaBoost	n_estimators=50	learning_rate=1.0	loss='linear'	1705.48	1724.645	↑19.16
AdaBoost	n_estimators=50	learning_rate=0.1	loss='linear'	1463.55	1464.4	↑0.843
AdaBoost	n_estimators=100	learning_rate=0.1	loss='linear'	1491.76	1493.297	↑1.533
AdaBoost	n_estimators=50	learning_rate=0.1	loss='square'	1489.18	1482.03	↓7.15
AdaBoost	n_estimators=10	learning_rate=0.1	loss='linear'	1454.44	1450.237	↓4.208
Random Forest	n_estimators=10	max_depth=None	criterion='mse'	1276.87	1284.46	↑7.5859
Bagging	n_estimators=10			1276.92	1278.26	↑1.33
Bagging	n_estimators=100			1222.12	1222.2058	↑0.0077

We expected the MAE would decrease for each model when the two attributes were delected. But, only one model (AdaBoost with loss = 'square') shows the decreased MAE.

We think that cont11 with cont12, and cont1 with cont9 are all not very important for the prediction of 'loss'.

10. Contribution of team members

Xunde Wang

Xiangru Zhou

11. References

Wikipedia (https://www.wikipedia.org/)

Scikit-learn (http://scikit-learn.org/stable/)

Kaggle (https://www.kaggle.com/)