Allstate Claims Severity

Pujan Malavia

In [1]:
```
from IPython.display import display
from PIL import Image
path= "C:/Users/puj83/OneDrive/Portfolio/Allstate_Claims_Severity/allstate.jp
g"
display(Image.open(path))
```

Link to Dataset:

[https://www.kaggle.com/c/allstate-claims-severity/data (https://www.kaggle.com/c/allstate-claims-severity/data)](https://www.kaggle.com/c/allstate-claims-severity/data)

Abstract:

When you’ve been devastated by a serious car accident, your focus is on the things that matter the most: family, friends, and other loved ones. Pushing paper with your insurance agent is the last place you want your time or mental energy spent. This is why Allstate, a personal insurer in the United States, is continually seeking fresh ideas to improve their claims service for the over 16 million households they protect.

Allstate is currently developing automated methods of predicting the cost, and hence severity, of claims. In this recruitment challenge, Kagglers are invited to show off their creativity and flex their technical chops by creating an algorithm which accurately predicts claims severity. Aspiring competitors will demonstrate insight into better ways to predict claims severity for the chance to be part of Allstate’s efforts to ensure a worry-free customer experience.

Industry:

Insurance

Company Information:

We are the Good Hands: We help people realize their hopes and dreams through products and services designed to protect them from life’s uncertainties and to prepare them for the future.

The Allstate Corporation is the largest publicly held personal lines property and casualty insurer in America. Allstate was founded in 1931 and became a publicly traded company in 1993.

Allstate offers car insurance, home, property, condo and renters insurance, plus insurance for recreational vehicles like motorcycles, boats and more.

The Allstate family of companies offers financial products including college savings programs, retirement planning and a range of life insurance products including term life and whole life.

https://www.linkedin.com/company/allstate (https://www.linkedin.com/company/allstate)

https://www.allstate.com/ (https://www.allstate.com/)

Use Case:

Creating an algorithm which accurately predicts claims severity

Tool:

Python (Jupyter Notebook)

Initial Dataset(s):

train.csv - the training set

test.csv - the test set. You must predict the loss value for the ids in this file.

sample_submission.csv - a sample submission file in the correct format

Data:

Each row in this dataset represents an insurance claim. You must predict the value for the 'loss' column. Variables prefaced with 'cat' are categorical, while those prefaced with 'cont' are continuous.

Data Fields:

id

cat1

cat2 ......

cat115

cat116

cont1

cont2 ......

cont13

cont14

loss

Importing Libraries

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_auc_score as AUC
from sklearn.metrics import mean_absolute_error
from sklearn.decomposition import PCA
from sklearn.preprocessing import LabelEncoder, LabelBinarizer
# from sklearn.cross_validation import cross_val_score
from sklearn.model_selection import train_test_split

from scipy import stats
import seaborn as sns
from copy import deepcopy

%matplotlib inline

# This may raise an exception in earlier versions of Jupyter
%config InlineBackend.figure_format = 'retina'
```

Importing Dataset(s)

In [3]:
```
train = pd.read_csv('C:/Users/puj83/OneDrive/Portfolio/\Allstate_Claims_Severi
ty/train.csv')
test = pd.read_csv('C:/Users/puj83/OneDrive/Portfolio/\Allstate_Claims_Severit
y/test.csv')
```

In [4]:
```
train.shape # Printing out dimensions of train dataset
```

Out[4]:
```
(188318, 132)
```

In [5]:
```
test.shape # Printing out dimensions of test dataset
```

Out[5]:
```
(125546, 131)
```

In [6]:
```
train.describe #
```

Out[6]:
```
<bound method NDFrame.describe of          id cat1 cat2 cat3 cat4 cat5 cat
6 cat7 cat8 cat9   ...    \
0            1    A    B    A    B    A    A    A    A    B   ...
1            2    A    B    A    A    A    A    A    A    B   ...
2            5    A    B    A    A    B    A    A    A    B   ...
3           10    B    B    A    B    A    A    A    A    B   ...
4           11    A    B    A    B    A    A    A    A    B   ...
5           13    A    B    A    A    A    A    A    A    B   ...
6           14    A    A    A    A    B    A    A    A    A   ...
7           20    A    B    A    B    A    A    A    A    B   ...
8           23    A    B    B    B    B    A    A    A    B   ...
9           24    A    B    A    A    B    B    A    A    B   ...
10          25    A    B    A    A    A    A    A    A    B   ...
11          33    A    B    A    A    B    A    A    A    B   ...
12          34    B    A    A    A    B    A    A    A    A   ...
13          41    B    A    A    A    B    B    A    A    A   ...
14          47    A    A    A    A    B    A    A    A    A   ...
15          48    A    A    A    A    B    B    A    A    A   ...
16          49    A    B    B    A    A    A    A    A    B   ...
17          51    A    A    A    A    A    B    A    A    A   ...
18          52    A    A    B    A    A    B    A    A    A   ...
19          55    A    A    A    B    A    A    A    A    A   ...
20          57    B    B    A    B    A    A    A    A    B   ...
21          60    A    A    A    B    A    B    A    A    A   ...
22          61    B    A    A    A    B    B    A    A    A   ...
23          66    B    A    A    B    A    A    A    A    A   ...
24          73    B    A    A    A    A    A    A    A    A   ...
25          76    A    A    A    B    A    A    A    A    A   ...
26          86    A    A    A    A    A    B    A    A    A   ...
27          89    B    A    A    B    A    A    A    A    A   ...
28          90    A    B    A    B    A    B    A    A    B   ...
29          93    A    A    A    A    B    A    A    A    A   ...
...        ...  ...  ...  ...  ...  ...  ...  ...  ...  ...   ...
188288  587563    A    A    A    B    A    A    A    A    A   ...
188289  587564    A    A    A    A    A    B    A    A    A   ...
188290  587566    A    A    A    B    A    A    A    A    A   ...
188291  587567    B    A    A    A    B    A    A    A    A   ...
188292  587569    B    A    A    A    A    B    A    A    A   ...
188293  587570    A    A    A    B    A    A    A    A    A   ...
188294  587572    A    A    A    B    A    B    A    A    A   ...
188295  587573    A    B    A    A    A    A    A    A    B   ...
188296  587574    A    A    A    A    B    A    B    A    A   ...
188297  587575    A    B    A    A    A    A    A    A    B   ...
188298  587578    A    A    B    A    A    B    A    A    A   ...
188299  587579    A    A    A    A    B    A    A    A    A   ...
188300  587580    A    A    A    B    A    A    A    A    A   ...
188301  587584    A    A    A    A    B    A    A    A    A   ...
188302  587592    A    A    A    B    B    A    A    A    A   ...
188303  587595    A    B    A    B    A    A    A    A    B   ...
188304  587601    A    A    A    A    B    A    A    B    A   ...
188305  587602    A    A    A    A    A    B    A    A    A   ...
188306  587603    A    B    A    A    B    A    A    A    B   ...
188307  587605    B    A    A    A    A    B    A    A    A   ...
188308  587606    A    A    A    A    B    A    A    A    A   ...
188309  587607    A    B    A    B    B    B    A    A    B   ...
188310  587611    A    B    A    A    B    A    A    A    B   ...
188311  587612    A    A    A    A    B    A    A    A    A   ...
```

```
188312  587619    A    A    A    A    A    B    A    A    A    ...
188313  587620    A    B    A    A    A    A    A    A    B    ...
188314  587624    A    A    A    A    A    B    A    A    A    ...
188315  587630    A    B    A    A    A    A    A    B    B    ...
188316  587632    A    B    A    A    A    A    A    A    B    ...
188317  587633    B    A    A    B    A    A    A    A    A    ...

           cont6     cont7    cont8    cont9   cont10    cont11    cont12  \
0       0.718367  0.335060  0.30260  0.67135  0.83510  0.569745  0.594646
1       0.438917  0.436585  0.60087  0.35127  0.43919  0.338312  0.366307
2       0.289648  0.315545  0.27320  0.26076  0.32446  0.381398  0.373424
3       0.440945  0.391128  0.31796  0.32128  0.44467  0.327915  0.321570
4       0.178193  0.247408  0.24564  0.22089  0.21230  0.204687  0.202213
5       0.364464  0.401162  0.26847  0.46226  0.50556  0.366788  0.359249
6       0.381515  0.363768  0.24564  0.40455  0.47225  0.334828  0.352251
7       0.867021  0.583389  0.90267  0.84847  0.80218  0.644013  0.785706
8       0.628534  0.384099  0.61229  0.38249  0.51111  0.682315  0.669033
9       0.713343  0.469223  0.30260  0.67135  0.83510  0.863052  0.879347
10      0.429383  0.877905  0.39455  0.53565  0.50556  0.550529  0.538473
11      0.314683  0.370419  0.58354  0.46226  0.38016  0.644013  0.665644
12      0.408772  0.363312  0.32843  0.32128  0.44467  0.327915  0.321570
13      0.241574  0.255339  0.58934  0.32496  0.26029  0.257148  0.253044
14      0.894903  0.586433  0.80058  0.93383  0.78770  0.880469  0.871011
15      0.570733  0.547756  0.80438  0.44352  0.63026  0.385085  0.377003
16      0.411902  0.593548  0.31796  0.38846  0.48889  0.457203  0.447145
17      0.688705  0.437192  0.67263  0.83505  0.59334  0.678924  0.665644
18      0.443265  0.637086  0.36636  0.52938  0.39068  0.678924  0.665644
19      0.436312  0.544355  0.48864  0.36285  0.20496  0.388786  0.406090
20      0.441525  0.437192  0.31796  0.32128  0.44467  0.377724  0.369858
21      0.349885  0.381185  0.81542  0.32311  0.36458  0.453334  0.454705
22      0.183243  0.253560  0.40028  0.21374  0.19431  0.167024  0.165648
23      0.373500  0.381883  0.36083  0.44352  0.45017  0.338312  0.366307
24      0.382070  0.451203  0.33906  0.47900  0.54433  0.812519  0.800726
25      0.592478  0.496452  0.29758  0.46226  0.51111  0.434083  0.424625
26      0.435733  0.769905  0.60087  0.40252  0.28677  0.550529  0.538473
27      0.373500  0.356037  0.36083  0.44352  0.45017  0.291268  0.295524
28      0.671307  0.464924  0.33906  0.62542  0.66076  0.607500  0.594646
29      0.557431  0.402942  0.34445  0.52728  0.79139  0.377724  0.369858
...          ...       ...      ...      ...      ...       ...       ...
188288  0.482425  0.414750  0.67263  0.51890  0.60401  0.464956  0.454705
188289  0.690216  0.498919  0.33906  0.62542  0.73106  0.622276  0.609277
188290  0.688705  0.490407  0.33906  0.62542  0.73106  0.622276  0.609277
188291  0.808048  0.694312  0.94145  0.64103  0.80218  0.745820  0.753252
188292  0.484775  0.480521  0.28768  0.42289  0.46119  0.430255  0.420899
188293  0.850938  0.611159  0.68823  0.91644  0.83510  0.569745  0.576121
188294  0.197932  0.314927  0.41762  0.26401  0.23545  0.207238  0.204687
188295  0.651024  0.452181  0.33906  0.62542  0.69471  0.492200  0.481306
188296  0.625784  0.606340  0.51256  0.42084  0.57172  0.665172  0.651918
188297  0.448496  0.735978  0.36083  0.40657  0.40666  0.776962  0.800726
188298  0.415039  0.395131  0.24123  0.32865  0.40666  0.352419  0.345316
188299  0.563226  0.451570  0.54829  0.29618  0.36974  0.472726  0.462286
188300  0.835720  0.794598  0.53046  0.50840  0.67554  0.742852  0.729856
188301  0.425928  0.636286  0.27797  0.50420  0.31003  0.742852  0.780521
188302  0.349083  0.368005  0.41762  0.41675  0.39068  0.275431  0.270746
188303  0.806951  0.555567  0.74629  0.93383  0.78770  0.757468  0.772574
188304  0.437758  0.535749  0.54236  0.47900  0.51111  0.705501  0.692256
188305  0.674671  0.699628  0.30768  0.38249  0.69471  0.607500  0.594646
```

188306	0.728484	0.414750	0.30260	0.67135	0.83510	0.872013	0.879347
188307	0.599275	0.548122	0.48864	0.45391	0.64056	0.592525	0.590961
188308	0.201125	0.259395	0.24564	0.30859	0.21983	0.207238	0.204687
188309	0.269520	0.338963	0.33906	0.28066	0.30529	0.245410	0.261799
188310	0.186254	0.317274	0.27797	0.32128	0.24355	0.180456	0.178698
188311	0.502705	0.473897	0.43518	0.66201	0.58257	0.415029	0.406090
188312	0.445008	0.377930	0.36636	0.29095	0.44467	0.327915	0.321570
188313	0.242437	0.289949	0.24564	0.30859	0.32935	0.223038	0.220003
188314	0.334270	0.382000	0.63475	0.40455	0.47779	0.307628	0.301921
188315	0.345883	0.370534	0.24564	0.45808	0.47779	0.445614	0.443374
188316	0.704364	0.562866	0.34987	0.44767	0.53881	0.863052	0.852865
188317	0.844563	0.533048	0.97123	0.93383	0.83814	0.932195	0.946432

	cont13	cont14	loss
0	0.822493	0.714843	2213.18
1	0.611431	0.304496	1283.60
2	0.195709	0.774425	3005.09
3	0.605077	0.602642	939.85
4	0.246011	0.432606	2763.85
5	0.345247	0.726792	5142.87
6	0.342239	0.382931	1132.22
7	0.859764	0.242416	3585.75
8	0.756454	0.361191	10280.20
9	0.822493	0.294523	6184.59
10	0.336261	0.715009	6396.85
11	0.339244	0.799124	5965.73
12	0.605077	0.818358	1193.05
13	0.276878	0.477578	1071.77
14	0.822493	0.251278	585.18
15	0.516660	0.340325	1395.45
16	0.301535	0.205651	6609.32
17	0.684242	0.407411	2658.70
18	0.304350	0.310796	4167.32
19	0.648701	0.830931	3797.89
20	0.605077	0.743810	1155.48
21	0.651733	0.354002	891.14
22	0.404520	0.725941	765.97
23	0.339244	0.793518	771.58
24	0.246011	0.215055	7256.49
25	0.357400	0.311644	1528.73
26	0.298734	0.698006	4787.07
27	0.339244	0.804795	2163.97
28	0.678452	0.285224	11673.03
29	0.687115	0.297788	1753.50
...	...	...	...
188288	0.407736	0.675983	2384.79
188289	0.687115	0.360712	961.10
188290	0.687115	0.342155	2786.15
188291	0.717751	0.216113	2157.66
188292	0.282249	0.238973	644.29
188293	0.828258	0.243950	4301.82
188294	0.271571	0.813596	4446.20
188295	0.678452	0.382540	1996.00
188296	0.614594	0.836524	16569.90
188297	0.287682	0.804795	4620.56
188298	0.624025	0.290736	3201.50
188299	0.657761	0.239309	1946.11

```
188300  0.663739  0.804769    839.41
188301  0.333292  0.359434    896.57
188302  0.256038  0.313505   1667.38
188303  0.812550  0.843080   4003.79
188304  0.357400  0.283936  12065.38
188305  0.684242  0.383437   4958.36
188306  0.833874  0.708475   2594.72
188307  0.701266  0.362479   1173.30
188308  0.357400  0.348217   2161.12
188309  0.181433  0.398571   4080.42
188310  0.304350  0.381660   4659.57
188311  0.354344  0.377315    994.85
188312  0.731059  0.721499    804.28
188313  0.333292  0.208216   1198.62
188314  0.318646  0.305872   1108.34
188315  0.339244  0.503888   5762.64
188316  0.654753  0.721707   1562.87
188317  0.810511  0.721460   4751.72

[188318 rows x 132 columns]>
```

In [7]:
```
test.describe
```

Out[7]:
```
<bound method NDFrame.describe of          id cat1 cat2 cat3 cat4 cat5 cat
6 cat7 cat8 cat9    ...     \
0            4    A    B    A    A    A    A    A    A    B    ...
1            6    A    B    A    B    A    A    A    A    B    ...
2            9    A    B    A    B    B    A    B    A    B    ...
3           12    A    A    A    A    B    A    A    A    A    ...
4           15    B    A    A    A    A    B    A    A    A    ...
5           17    A    A    A    A    B    A    A    A    A    ...
6           21    B    A    A    A    B    B    A    A    A    ...
7           28    B    B    A    A    A    A    A    A    B    ...
8           32    A    B    A    A    A    A    A    A    B    ...
9           43    A    B    A    A    A    A    A    A    B    ...
10          46    A    A    A    A    A    B    A    A    A    ...
11          50    A    A    A    A    B    B    A    A    A    ...
12          54    B    A    A    A    B    A    A    A    A    ...
13          62    B    A    A    A    A    B    A    A    A    ...
14          70    A    A    A    A    B    B    A    A    A    ...
15          71    A    A    A    A    A    B    A    A    A    ...
16          75    A    A    A    A    B    A    A    A    A    ...
17          77    A    A    A    B    A    B    A    A    A    ...
18          81    A    A    A    A    B    A    A    A    A    ...
19          83    A    B    A    B    A    A    A    A    B    ...
20          87    A    A    A    A    B    A    A    A    A    ...
21          97    A    A    A    B    A    A    A    A    A    ...
22         103    A    B    A    A    A    A    A    A    B    ...
23         119    B    A    A    A    A    B    A    A    A    ...
24         120    A    A    A    B    A    A    A    A    A    ...
25         127    A    A    A    A    A    B    A    A    A    ...
26         138    A    B    A    B    A    A    A    A    B    ...
27         141    A    A    A    A    B    B    A    A    A    ...
28         148    A    A    A    A    B    A    A    A    A    ...
29         150    B    B    A    A    A    A    A    A    B    ...
...        ...  ...  ...  ...  ...  ...  ...  ...  ...  ...    ...
125516  587482    A    B    A    A    A    A    A    B    B    ...
125517  587484    A    B    A    B    B    A    A    A    B    ...
125518  587489    A    B    A    A    B    A    A    A    B    ...
125519  587494    B    A    A    A    A    B    A    A    A    ...
125520  587509    B    B    A    A    A    A    A    A    B    ...
125521  587511    A    A    A    A    A    B    A    A    A    ...
125522  587515    A    A    A    A    A    B    A    A    A    ...
125523  587517    A    B    A    A    A    B    A    A    B    ...
125524  587519    A    A    A    A    A    B    A    A    A    ...
125525  587524    A    A    A    A    B    A    A    A    A    ...
125526  587531    A    A    A    A    A    B    A    A    A    ...
125527  587532    A    A    A    B    B    A    A    A    A    ...
125528  587534    A    A    A    A    A    B    A    A    A    ...
125529  587538    A    A    B    A    A    A    A    A    A    ...
125530  587540    A    B    A    A    B    A    A    A    B    ...
125531  587548    A    B    A    A    A    A    A    A    B    ...
125532  587549    A    A    A    B    B    A    B    A    A    ...
125533  587560    A    A    A    B    A    A    A    A    A    ...
125534  587561    A    B    A    A    B    A    A    A    B    ...
125535  587581    B    A    A    A    B    B    A    A    A    ...
125536  587583    B    A    A    A    B    A    A    A    A    ...
125537  587587    A    A    A    B    A    A    A    A    A    ...
125538  587596    A    B    A    A    B    A    A    A    B    ...
125539  587610    A    A    A    B    A    A    A    A    A    ...
```

125540	587613	A	A	B	A	A	B	A	A	A	...
125541	587617	A	A	A	B	A	A	A	A	A	...
125542	587621	A	A	A	A	B	B	A	B	A	...
125543	587627	B	B	A	A	B	A	A	A	B	...
125544	587629	A	A	A	A	A	B	A	B	A	...
125545	587634	A	B	A	A	A	A	A	A	B	...

	cont5	cont6	cont7	cont8	cont9	cont10	cont11	\
0	0.281143	0.466591	0.317681	0.61229	0.34365	0.38016	0.377724	
1	0.836443	0.482425	0.443760	0.71330	0.51890	0.60401	0.689039	
2	0.718531	0.212308	0.325779	0.29758	0.34365	0.30529	0.245410	
3	0.397069	0.369930	0.342355	0.40028	0.33237	0.31480	0.348867	
4	0.302678	0.398862	0.391833	0.23688	0.43731	0.50556	0.359572	
5	0.643315	0.407351	0.390540	0.46477	0.46853	0.50556	0.607500	
6	0.281143	0.960845	0.740081	0.75964	0.98330	0.82249	0.863052	
7	0.651246	0.451115	0.316313	0.27320	0.52100	0.50556	0.415029	
8	0.534484	0.343492	0.358758	0.81900	0.32128	0.36458	0.453334	
9	0.281143	0.394921	0.287416	0.92347	0.48320	0.24766	0.359572	
10	0.405415	0.457821	0.774678	0.33372	0.41471	0.47779	0.760322	
11	0.696981	0.627435	0.451447	0.52450	0.64873	0.79139	0.472726	
12	0.281143	0.644854	0.724803	0.34987	0.48320	0.67065	0.695685	
13	0.867056	0.364464	0.401162	0.26847	0.46226	0.50556	0.366788	
14	0.281143	0.636189	0.793953	0.37754	0.41471	0.58796	0.705501	
15	0.281143	0.677761	0.492874	0.32317	0.68419	0.72223	0.441763	
16	0.281143	0.522095	0.694731	0.42930	0.53774	0.61459	0.472726	
17	0.281143	0.671047	0.478451	0.79674	0.76280	0.45567	0.480509	
18	0.281143	0.464250	0.573884	0.50658	0.39849	0.57172	0.810130	
19	0.491114	0.334009	0.302292	0.31280	0.39648	0.38016	0.341813	
20	0.939556	0.242437	0.289949	0.24564	0.30859	0.32935	0.223038	
21	0.356315	0.236017	0.271089	0.36083	0.30859	0.23948	0.180456	
22	0.310061	0.510054	0.465660	0.29260	0.38249	0.58796	0.730752	
23	0.725503	0.825463	0.635715	0.66201	0.64681	0.68039	0.888438	
24	0.517162	0.755048	0.659806	0.34987	0.55855	0.69471	0.909611	
25	0.718531	0.678275	0.556420	0.33906	0.62542	0.81255	0.636828	
26	0.491114	0.937400	0.608811	0.47669	0.97621	0.83510	0.511698	
27	0.281143	0.898390	0.600908	0.96769	0.85380	0.83814	0.841478	
28	0.911073	0.240069	0.279895	0.24564	0.30859	0.32446	0.223038	
29	0.456483	0.711657	0.755984	0.30768	0.38249	0.72667	0.654669	
...	...	...	...	...	...	...	...	
125516	0.718531	0.698205	0.526532	0.35533	0.38249	0.73971	0.607500	
125517	0.295397	0.862482	0.458423	0.29260	0.89888	0.81591	0.550529	
125518	0.281143	0.340845	0.347485	0.31280	0.39849	0.41743	0.314313	
125519	0.295397	0.458113	0.386882	0.36083	0.46226	0.36458	0.388786	
125520	0.499798	0.350956	0.363768	0.58354	0.44352	0.39599	0.341813	
125521	0.811271	0.688705	0.480915	0.33906	0.62542	0.73106	0.661688	
125522	0.551723	0.535867	0.907969	0.45883	0.52309	0.52221	0.705501	
125523	0.281143	0.421908	0.637770	0.34987	0.40657	0.40666	0.468839	
125524	0.380560	0.646468	0.411042	0.52450	0.64873	0.79139	0.377724	
125525	0.281143	0.271144	0.302709	0.41762	0.37065	0.38541	0.231253	
125526	0.718531	0.364464	0.401162	0.26847	0.46226	0.50556	0.415029	
125527	0.372405	0.487419	0.394895	0.93736	0.51050	0.43373	0.396226	
125528	0.281143	0.867697	0.724409	0.68823	0.71934	0.79863	0.837272	
125529	0.281143	0.431689	0.649542	0.33906	0.40455	0.47779	0.689039	
125530	0.674529	0.356602	0.338367	0.32843	0.32128	0.36974	0.307628	
125531	0.577339	0.571597	0.538203	0.45883	0.49370	0.52775	0.742852	
125532	0.380560	0.767863	0.669049	0.94012	0.64103	0.80218	0.745820	
125533	0.281143	0.731035	0.994883	0.64577	0.71934	0.62507	0.909611	

125534	0.805895	0.435733	0.453404	0.69840	0.39447	0.46119	0.430255
125535	0.281143	0.393798	0.533539	0.27797	0.50420	0.31003	0.678924
125536	0.281143	0.730804	0.994421	0.64577	0.71764	0.62507	0.909611
125537	0.534484	0.597862	0.475866	0.57187	0.73271	0.75234	0.588753
125538	0.281143	0.438917	0.705814	0.77668	0.35127	0.30060	0.569745
125539	0.310061	0.189484	0.265894	0.25918	0.24180	0.21230	0.169206
125540	0.931165	0.375429	0.389249	0.41182	0.42289	0.45017	0.341813
125541	0.281143	0.438917	0.815941	0.39455	0.48740	0.40666	0.550529
125542	0.674529	0.346948	0.424968	0.47669	0.25753	0.26894	0.324486
125543	0.794794	0.808958	0.511502	0.72299	0.94438	0.83510	0.933174
125544	0.302678	0.372125	0.388545	0.31796	0.32128	0.36974	0.307628
125545	0.413817	0.221699	0.242044	0.25461	0.31399	0.25183	0.245410

	cont12	cont13	cont14
0	0.369858	0.704052	0.392562
1	0.675759	0.453468	0.208045
2	0.241676	0.258586	0.297232
3	0.341872	0.592264	0.555955
4	0.352251	0.301535	0.825823
5	0.594646	0.250991	0.283976
6	0.879347	0.888944	0.787807
7	0.481306	0.199940	0.450597
8	0.443374	0.695650	0.295075
9	0.352251	0.519989	0.602666
10	0.747533	0.304350	0.305920
11	0.500382	0.689974	0.307258
12	0.682413	0.642600	0.838149
13	0.359249	0.345247	0.725605
14	0.698722	0.611431	0.822254
15	0.432101	0.689974	0.729392
16	0.462286	0.324464	0.583785
17	0.481306	0.723122	0.202501
18	0.798279	0.279556	0.652859
19	0.352251	0.261150	0.599018
20	0.220003	0.333292	0.818011
21	0.443374	0.236253	0.811269
22	0.717648	0.642600	0.195278
23	0.879347	0.802184	0.814328
24	0.901612	0.666708	0.234711
25	0.623714	0.675536	0.381660
26	0.576121	0.919827	0.475459
27	0.909444	0.856518	0.599415
28	0.220003	0.333292	0.806101
29	0.641454	0.684242	0.714791
...	...	...	...
125516	0.594646	0.736269	0.721896
125517	0.623714	0.919827	0.602363
125518	0.308395	0.351299	0.254988
125519	0.380595	0.648701	0.719271
125520	0.352251	0.339244	0.236616
125521	0.648446	0.687115	0.357316
125522	0.692256	0.330336	0.804035
125523	0.458493	0.287682	0.807022
125524	0.369858	0.689974	0.838158
125525	0.241676	0.388569	0.390576
125526	0.406090	0.345247	0.230681
125527	0.387819	0.633362	0.723703

```
125528  0.826178  0.879390  0.624095
125529  0.675759  0.315758  0.725515
125530  0.301921  0.608259  0.812106
125531  0.729856  0.369740  0.728514
125532  0.753252  0.717751  0.235890
125533  0.901612  0.354344  0.294556
125534  0.519456  0.605077  0.776205
125535  0.729856  0.333292  0.772099
125536  0.901612  0.354344  0.818373
125537  0.576121  0.768525  0.601881
125538  0.557380  0.274217  0.387062
125539  0.167768  0.339244  0.833053
125540  0.335036  0.382252  0.836701
125541  0.538473  0.298734  0.345946
125542  0.352251  0.490001  0.290576
125543  0.926619  0.848129  0.808125
125544  0.301921  0.608259  0.361542
125545  0.241676  0.287682  0.220323

[125546 rows x 131 columns]>
```

In [8]:
```
print ('First 20 columns:', list(train.columns[:20]))
print ('Last 20 columns:', list(train.columns[-20:]))
```

```
First 20 columns: ['id', 'cat1', 'cat2', 'cat3', 'cat4', 'cat5', 'cat6', 'cat
7', 'cat8', 'cat9', 'cat10', 'cat11', 'cat12', 'cat13', 'cat14', 'cat15', 'ca
t16', 'cat17', 'cat18', 'cat19']
Last 20 columns: ['cat112', 'cat113', 'cat114', 'cat115', 'cat116', 'cont1',
'cont2', 'cont3', 'cont4', 'cont5', 'cont6', 'cont7', 'cont8', 'cont9', 'cont
10', 'cont11', 'cont12', 'cont13', 'cont14', 'loss']
```

In [9]:
```
pd.isnull(train).values.any()
```

Out[9]:
```
False
```

In [10]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188318 entries, 0 to 188317
Columns: 132 entries, id to loss
dtypes: float64(15), int64(1), object(116)
memory usage: 189.7+ MB
```

In [11]:
```
cat_features = list(train.select_dtypes(include=['object']).columns)
print ("Categorical: {} features".format(len(cat_features)))
```

```
Categorical: 116 features
```

In [12]:
```
cont_features = [cont for cont in list(train.select_dtypes(
                 include=['float64', 'int64']).columns) if cont not in ['loss'
, 'id']]
print ("Continuous: {} features".format(len(cont_features)))
```

```
Continuous: 14 features
```

In [13]:
```
id_col = list(train.select_dtypes(include=['int64']).columns)
print ("A column of int64: {}".format(id_col))
```

```
A column of int64: ['id']
```

In [14]:
```
cat_uniques = []
for cat in cat_features:
    cat_uniques.append(len(train[cat].unique()))

uniq_values_in_categories = pd.DataFrame.from_items([('cat_name', cat_features
), ('unique_values', cat_uniques)])
```

```
C:\Users\puj83\anaconda3\lib\site-packages\ipykernel_launcher.py:5: FutureWar
ning: from_items is deprecated. Please use DataFrame.from_dict(dict(items),
...) instead. DataFrame.from_dict(OrderedDict(items)) may be used to preserve
the key order.
  """
```

In [15]:
```
uniq_values_in_categories.head()
```

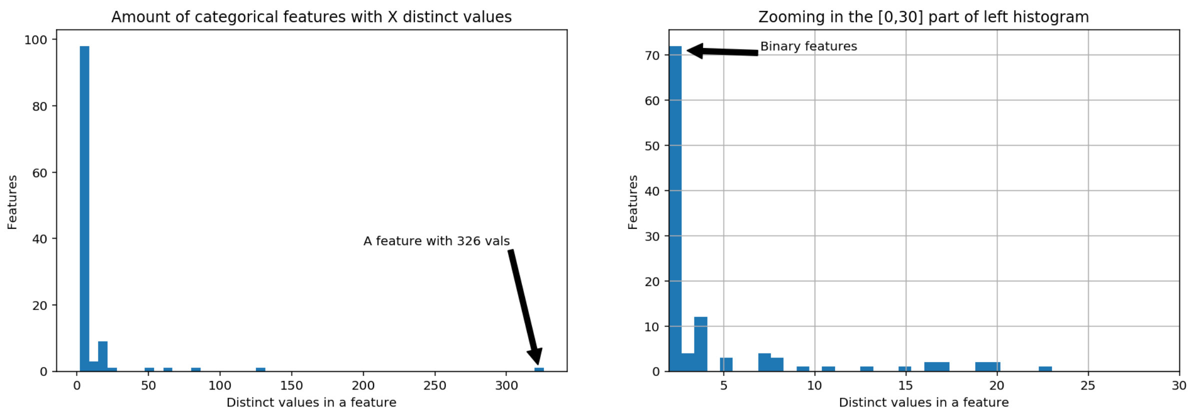
Out[15]:

	cat_name	unique_values
0	cat1	2
1	cat2	2
2	cat3	2
3	cat4	2
4	cat5	2

In [16]:
```
fig, (ax1, ax2) = plt.subplots(1,2)
fig.set_size_inches(16,5)
ax1.hist(uniq_values_in_categories.unique_values, bins=50)
ax1.set_title('Amount of categorical features with X distinct values')
ax1.set_xlabel('Distinct values in a feature')
ax1.set_ylabel('Features')
ax1.annotate('A feature with 326 vals', xy=(322, 2), xytext=(200, 38), arrowpr
ops=dict(facecolor='black'))

ax2.set_xlim(2,30)
ax2.set_title('Zooming in the [0,30] part of left histogram')
ax2.set_xlabel('Distinct values in a feature')
ax2.set_ylabel('Features')
ax2.grid(True)
ax2.hist(uniq_values_in_categories[uniq_values_in_categories.unique_values <=
30].unique_values, bins=30)
ax2.annotate('Binary features', xy=(3, 71), xytext=(7, 71), arrowprops=dict(fa
cecolor='black'))
```

Out[16]:
```
Text(7, 71, 'Binary features')
```



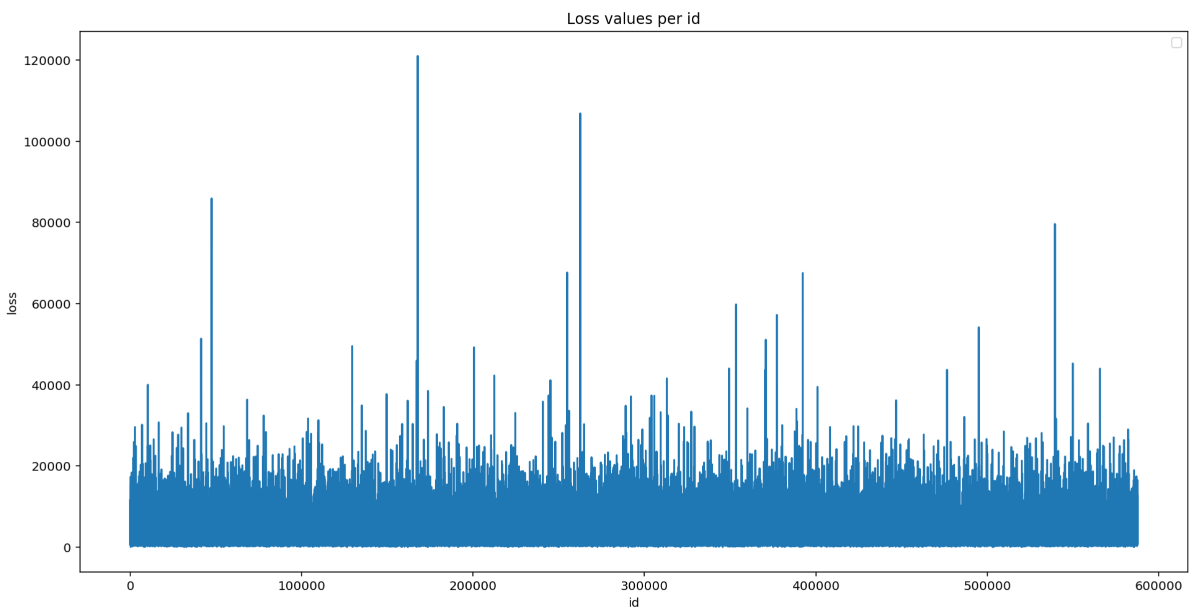
In [17]:
```
# Another option is to use Series.value_counts() method, but its
# output is not that nice

uniq_values = uniq_values_in_categories.groupby('unique_values').count()
uniq_values = uniq_values.rename(columns={'cat_name': 'categories'})
uniq_values.sort_values(by='categories', inplace=True, ascending=False)
uniq_values.reset_index(inplace=True)
print (uniq_values)
```

```
    unique_values  categories
0               2          72
1               4          12
2               3           4
3               7           4
4               5           3
5               8           3
6              20           2
7              19           2
8              17           2
9              16           2
10             15           1
11             13           1
12             11           1
13              9           1
14             23           1
15             51           1
16             61           1
17             84           1
18            131           1
19            326           1
```

In [18]:
```
plt.figure(figsize=(16,8))
plt.plot(train['id'], train['loss'])
plt.title('Loss values per id')
plt.xlabel('id')
plt.ylabel('loss')
plt.legend()
plt.show()
```

No handles with labels found to put in legend.



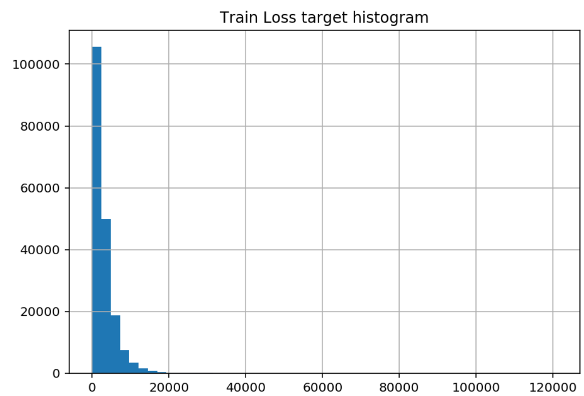
In [19]:
```
stats.mstats.skew(train['loss']).data
```

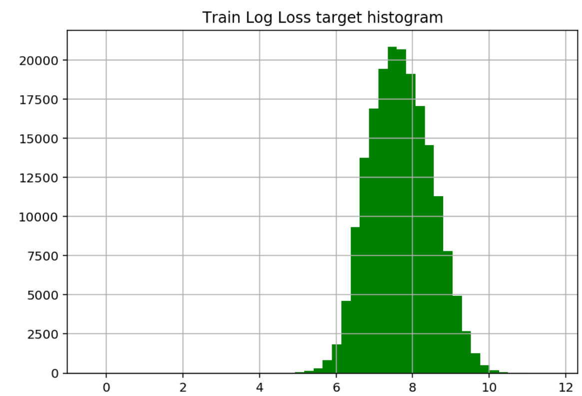
Out[19]: array(3.79492815)

In [20]:
```
stats.mstats.skew(np.log(train['loss'])).data
```

Out[20]: array(0.0929738)

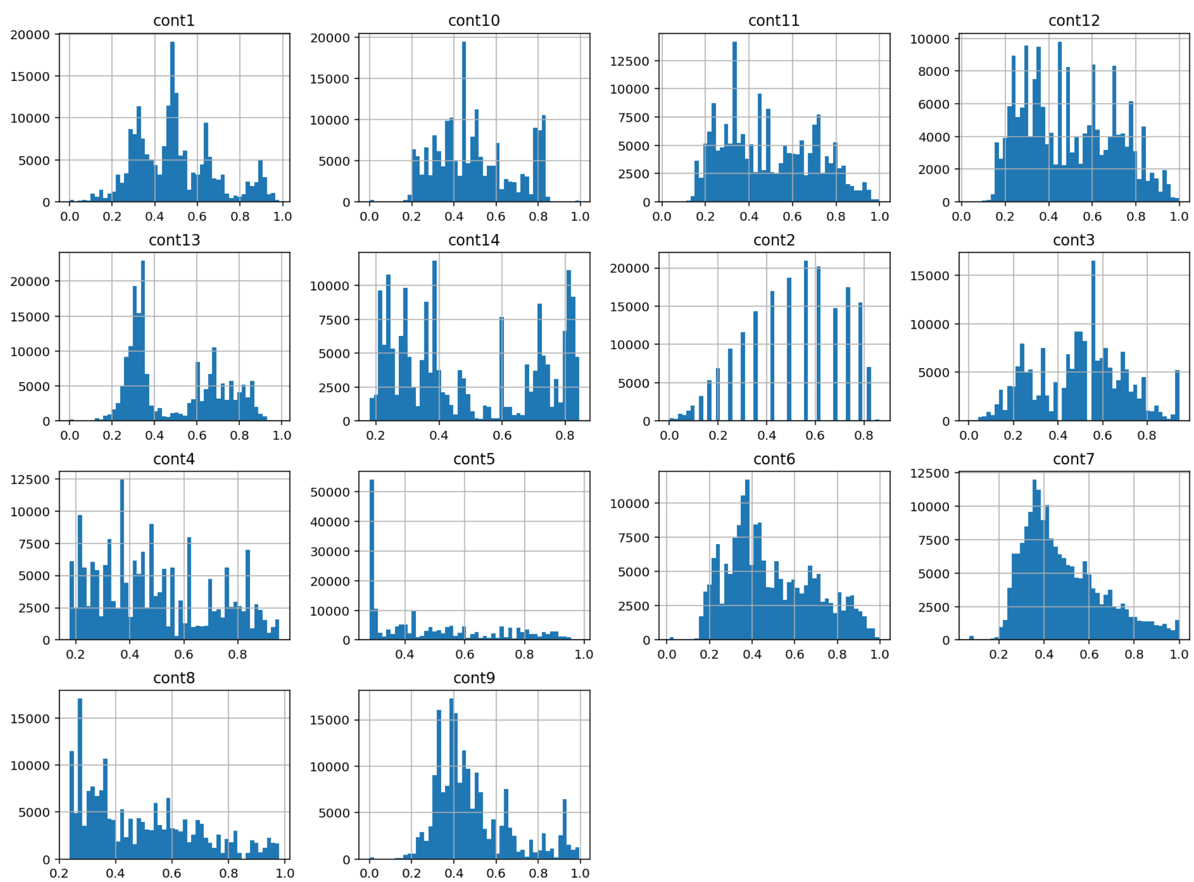
In [21]:
```
fig, (ax1, ax2) = plt.subplots(1,2)
fig.set_size_inches(16,5)
ax1.hist(train['loss'], bins=50)
ax1.set_title('Train Loss target histogram')
ax1.grid(True)
ax2.hist(np.log(train['loss']), bins=50, color='g')
ax2.set_title('Train Log Loss target histogram')
ax2.grid(True)
plt.show()
```





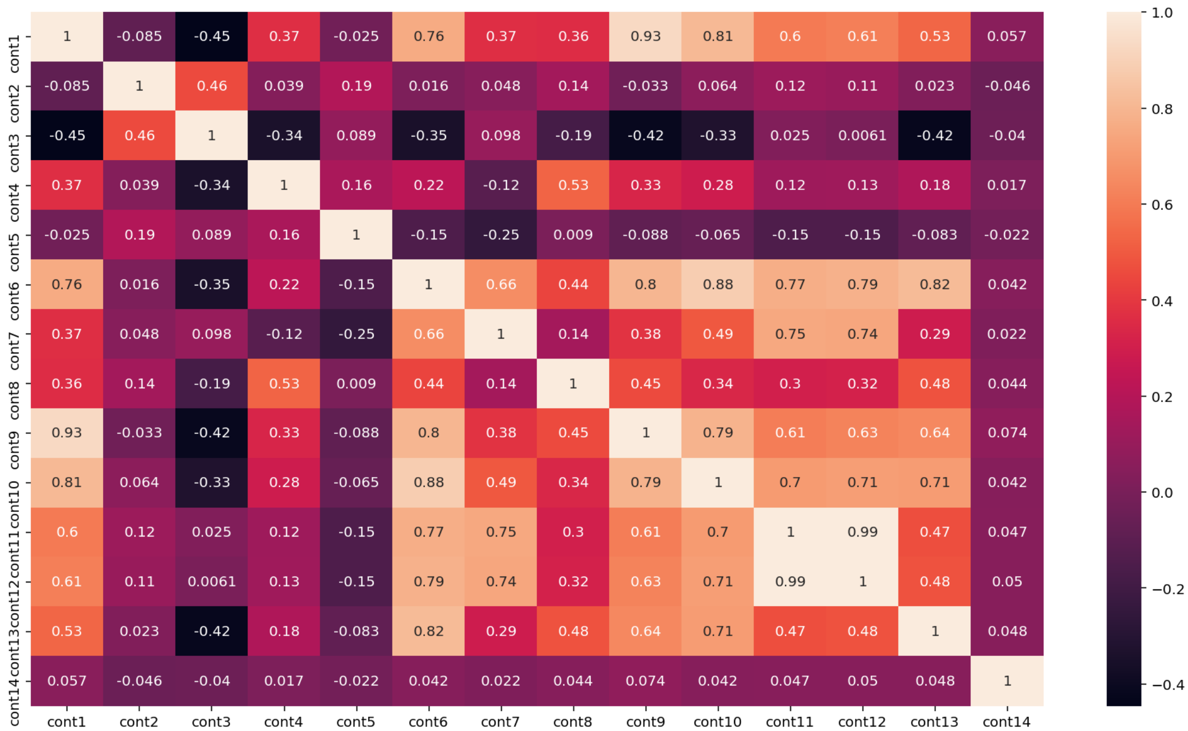
In [22]:
```
train[cont_features].hist(bins=50, figsize=(16,12))
```

Out[22]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001B180A30DC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180B6AEC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180BA5E88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180BD80C8
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001B180C0FA88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180C485C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180C806C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180CBA808
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001B180CC6408>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180CFF5C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180D63B48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180D9EBC8
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001B180DD7CC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B180E0DE08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B1813C8F08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001B181404108
>]],
      dtype=object)
```



In [23]:
```python
plt.subplots(figsize=(16,9))
correlation_mat = train[cont_features].corr()
sns.heatmap(correlation_mat, annot=True)
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1b182e850c8>
```



In [24]:
```python
# Simple data preparation

train_d = train.drop(['id','loss'], axis=1)
test_d = test.drop(['id'], axis=1)

# To make sure we can distinguish between two classes
train_d['Target'] = 1
test_d['Target'] = 0

# We concatenate train and test in one big dataset
data = pd.concat((train_d, test_d))

# We use label encoding for categorical features:
data_le = deepcopy(data)
for c in range(len(cat_features)):
    data_le[cat_features[c]] = data_le[cat_features[c]].astype('category').cat
.codes

# We use one-hot encoding for categorical features:
data = pd.get_dummies(data=data, columns=cat_features)
```

In [25]:
```
data = data.iloc[np.random.permutation(len(data))]
data.reset_index(drop = True, inplace = True)

x = data.drop(['Target'], axis = 1)
y = data.Target

train_examples = 100000

x_train = x[:train_examples]
x_test = x[train_examples:]
y_train = y[:train_examples]
y_test = y[train_examples:]
```

In [26]:
```
# Logistic Regression:
clf = LogisticRegression()
clf.fit(x_train, y_train)
pred = clf.predict_proba(x_test)[:,1]
auc = AUC(y_test, pred)
print ("Logistic Regression AUC: {:.2%}".format(auc))

# Random Forest, a simple model (100 trees) trained in parallel
clf = RandomForestClassifier(n_estimators=100, n_jobs=-1)
clf.fit(x_train, y_train)
pred = clf.predict_proba(x_test)[:,1]
auc = AUC(y_test, pred)
print ("Random Forest AUC: {:.2%}".format(auc))
```

```
C:\Users\puj83\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:
940: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)

Logistic Regression AUC: 50.05%
Random Forest AUC: 49.79%
```

In [ ]: