Allstate Kaggle Competition

Cristina Andronescu | Oamar Gianan | James Lee | Alex Rohr | Joseph van Bemmelen

Competition Background

How severe is an insurance claim?

> *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.*

Training data: 188,318 rows and 132 columns of unlabeled data
Test data: 125,546 rows and 131 columns of unlabeled data

Overview

- Exploring the data
- Preprocessing
- Supervised methods
  - Linear model
  - Ridge
  - Lasso
  - Random Forest
  - GBM
- Non-supervised methods
  - PCA
- Ensembling

Exploring the data

Training data: 188,318 rows and 132 columns of unlabeled data

- 72 binary categorical variables (2 levels)
- 43 non-binary categorical variables (3 to 326 levels)
- 14 continuous variables
- Continuous dependent variable “loss”

Test data: 125,546 rows and 131 columns of unlabeled data

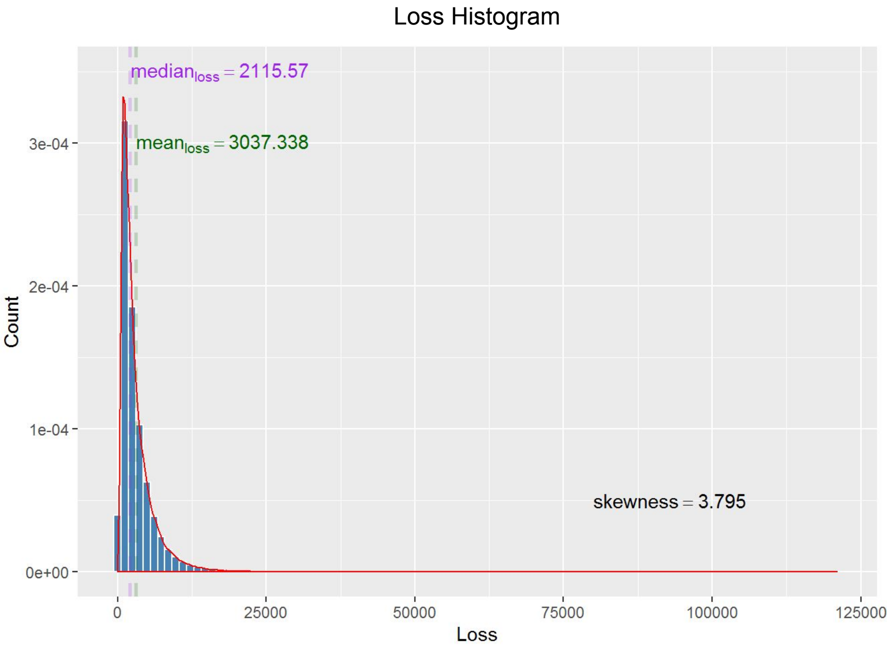
- Some of the variables have additional levels in the test set!

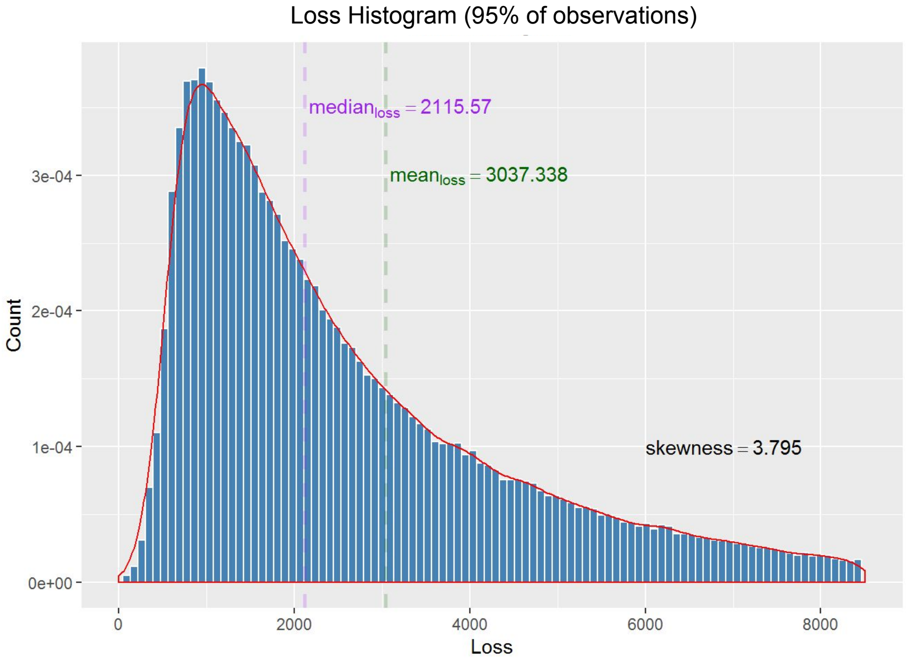
Exploring the data

```
dim(train)
str(train)
summary(train)
sapply(train, sd)
```

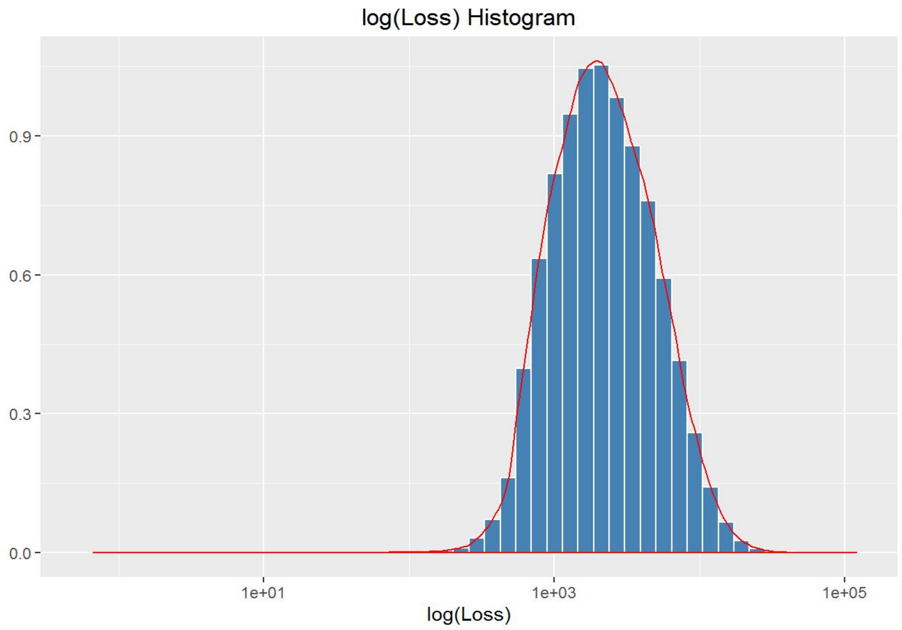
```
id           cat1        cat2        cat3        cat4        cat5        cat6        cat7        cat8
Min.   :     1   A:141550    A:106721    A:177993    A:128395    A:123737    A:131693    A:183744    A:177274
1st Qu.:147748   B: 46768    B: 81597    B: 10325    B: 59923    B: 64581    B: 56625    B:  4574    B: 11044
Median :294540
Mean   :294136
3rd Qu.:440681
Max.   :587633
…

    cat110           cat111           cat112           cat113           cat114           cat115
CL     :25305    A      :128395   E      :25148    BM     :26191    A      :131693   K      :43866
EG     :24654    C      : 32401   AH     :18639    AE     :22030    C      : 16793   O      :26813
CS     :24592    E      : 14682   AS     :17669    L      :13058    E      : 16475   J      :23895
EB     :21396    G      :  7039   J      :16222    AX     :12661    J      :  8199   N      :22438
CO     :17495    I      :  3578   AF     : 9368    Y      :11374    F      :  7905   P      :21538
BT     :16365    K      :  1353   AN     : 9138    K      : 7738    N      :  2455   L      :16125
(Other):58511    (Other):   870   (Other):92134    (Other):95266    (Other):  4798   (Other):33643

    cat116            cont1              cont2              cont3              cont4            cont5
HK     : 21061   Min.   :0.000016   Min.   :0.001149   Min.   :0.002634   Min.   :0.1769   Min.   :0.2811
DJ     : 20244   1st Qu.:0.346090   1st Qu.:0.358319   1st Qu.:0.336963   1st Qu.:0.3274   1st Qu.:0.2811
CK     : 10162   Median :0.475784   Median :0.555782   Median :0.527991   Median :0.4529   Median :0.4223
DP     :  9202   Mean   :0.493861   Mean   :0.507188   Mean   :0.498918   Mean   :0.4918   Mean   :0.4874
GS     :  8736   3rd Qu.:0.623912   3rd Qu.:0.681761   3rd Qu.:0.634224   3rd Qu.:0.6521   3rd Qu.:0.6433
CR     :  6862   Max.   :0.984975   Max.   :0.862654   Max.   :0.944251   Max.   :0.9543   Max.   :0.9837
(Other):112051

    cont6             cont7            cont8            cont9             cont10           cont11
Min.   :0.01268   Min.   :0.0695   Min.   :0.2369   Min.   :0.00008   Min.   :0.0000   Min.   :0.03532
1st Qu.:0.33610   1st Qu.:0.3502   1st Qu.:0.3128   1st Qu.:0.35897   1st Qu.:0.3646   1st Qu.:0.31096
Median :0.44094   Median :0.4383   Median :0.4411   Median :0.44145   Median :0.4612   Median :0.45720
Mean   :0.49094   Mean   :0.4850   Mean   :0.4864   Mean   :0.48551   Mean   :0.4981   Mean   :0.49351
3rd Qu.:0.65502   3rd Qu.:0.5910   3rd Qu.:0.6236   3rd Qu.:0.56682   3rd Qu.:0.6146   3rd Qu.:0.67892
Max.   :0.99716   Max.   :1.0000   Max.   :0.9802   Max.   :0.99540   Max.   :0.9950   Max.   :0.99874

    cont12            cont13             cont14            loss
Min.   :0.03623   Min.   :0.000228   Min.   :0.1797   Min.   :     0.67
1st Qu.:0.31166   1st Qu.:0.315758   1st Qu.:0.2946   1st Qu.:  1204.46
Median :0.46229   Median :0.363547   Median :0.4074   Median :  2115.57
Mean   :0.49315   Mean   :0.493138   Mean   :0.4957   Mean   :  3037.34
3rd Qu.:0.67576   3rd Qu.:0.689974   3rd Qu.:0.7246   3rd Qu.:  3864.05
Max.   :0.99848   Max.   :0.988494   Max.   :0.8448   Max.   :121012.25
```

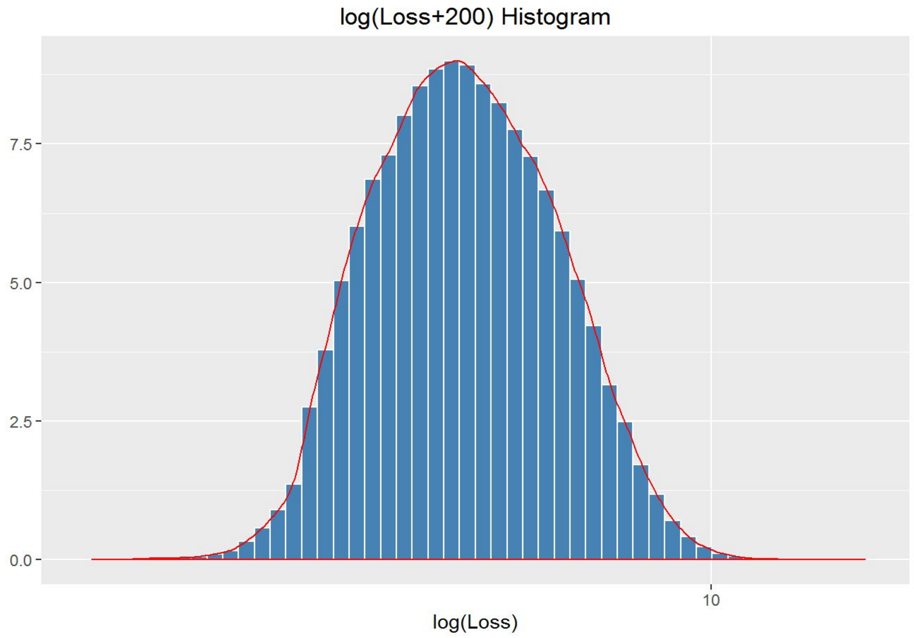
Exploring the data





Exploring the data





Preprocessing the data

Because of the many levels within the categorical variables, we will preprocess the data and create dummy columns for each level with values of 0 or 1.

In order to reduce the number of new columns, we will limit the dummy columns to categories that comprise at least 5% of the variable.

Additionally, we joined the raw train and test dataset to account for the levels that appear in the test.csv dataset, but not in the train.csv dataset.

Log transform was applied on the response column.

Preprocessing the data

```
library(caret)
library(mlbench)
library(Hmisc)
library(doMC)
registerDoMC(cores = 6)

##Reading the dataset
all.train <- read.csv("train.csv", row.names
= "id")
all.test <- read.csv("test.csv", row.names =
"id")

#make new train to combine into single model
to split later into train/test
all.train2 = all.train
all.train2$loss = NULL
all.test2 = rbind(all.test, all.train2)

##Converting categories to numeric
#this is done by first splitting the binary
level, multi-level, and
#continuous variables
bin.train <- all.test2[,1:72]
cat.train <- all.test2[,73:116]
cont.train <- all.test2[,117:130]
##Combine levels
#combining multiple levels using
combine.levels
#minimum 5%
#unique(bin.train$cat7)
# table(cat.train$cat100)
# unique(combine.levels(cat.train$cat100))
test <- sapply(cat.train, combine.levels)
test <- as.data.frame(test)

#cbind binary and reduced categorical levels
comb.train <- cbind(bin.train, test)
##Dummify all factor variables
dmy <- dummyVars(" ~ .", data = comb.train,
fullRank=T)
test <- as.data.frame(predict(dmy, newdata =
comb.train))
dim(test)
###writing to file
#write.csv(test, "comb_dum_train.csv")
##Combine dummified with cont vars
all.cd.train <- cbind(test, cont.train)
dim(all.cd.train)
#split dataset into new train and new test
with combine
new.all.cd.test = all.cd.train[1:125546,]
new.all.cd.train =
all.cd.train[125547:313864,]

#log transformation
#all.cd.train$loss <- log(all.cd.train$loss
+ 200)

#add log loss values to train set
new.all.cd.train$loss = log(all.train$loss
+200)
```

Preprocessing the data

```
> str(new.all.cd.train, list.len = 1000)

> str(new.all.cd.train, list.len = 1000)
'data.frame':	188318 obs. of  219 variables:
 $ cat1.B      : num  0 0 0 1 0 0 0 0 0 0 ...
 $ cat2.B      : num  1 1 1 1 1 1 0 1 1 1 ...
 $ cat3.B      : num  0 0 0 0 0 0 0 0 1 0 ...
 $ cat4.B      : num  1 0 0 1 1 0 0 1 1 0 ...
 $ cat5.B      : num  0 0 1 0 0 0 1 0 1 1 ...
 $ cat6.B      : num  0 0 0 0 0 0 0 0 0 1 ...
 $ cat7.B      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat8.B      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat9.B      : num  1 1 1 1 1 1 0 1 1 1 ...
 $ cat10.B     : num  0 1 1 0 1 0 0 0 1 0 ...
 $ cat11.B     : num  1 0 1 0 0 0 0 0 1 0 ...
 $ cat12.B     : num  0 0 1 0 1 0 0 0 1 0 ...
 $ cat13.B     : num  0 0 1 0 0 0 0 0 1 0 ...
 $ cat14.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat15.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat16.B     : num  0 0 0 0 0 0 0 0 1 0 ...
 $ cat17.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat18.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat19.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat20.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat21.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat22.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat23.B     : num  1 0 0 1 1 0 0 1 1 0 ...
 $ cat24.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat25.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat26.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat27.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat28.B     : num  0 0 0 0 0 0 0 1 0 0 ...
 $ cat29.B     : num  0 0 0 0 0 0 0 0 0 0 …
 $ cat30.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat31.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat32.B     : num  0 0 0 0 0 0 0 1 0 0 ...
```

```
 $ cat33.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat34.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat35.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat36.B     : num  0 0 1 0 0 0 1 0 1 1 ...
 $ cat37.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat38.B     : num  0 0 0 0 0 0 0 0 1 1 ...
 $ cat39.B     : num  0 0 0 0 0 0 0 0 0 1 ...
 $ cat40.B     : num  0 0 0 0 0 0 0 0 0 1 ...
 $ cat41.B     : num  0 0 0 0 0 0 1 0 0 0 ...
 $ cat42.B     : num  0 0 0 0 0 0 0 0 0 1 ...
 $ cat43.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat44.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat45.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat46.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat47.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat48.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat49.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat50.B     : num  0 0 0 0 0 0 0 0 0 1 ...
 $ cat51.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat52.B     : num  0 0 0 0 0 0 0 0 0 1 ...
 $ cat53.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat54.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat55.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat56.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat57.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat58.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat59.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat60.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat61.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat62.B     : num  0 0 0 0 0 0 0 0 0 0 …
 $ cat63.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat64.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat65.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat66.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat67.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat68.B     : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
 $ cat69.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat70.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat71.B     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat72.B     : num  0 0 0 0 1 1 0 0 1 0 ...
 $ cat73.OTHER : num  0 0 0 1 0 0 0 0 0 0 ...
 $ cat74.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat75.OTHER : num  1 0 0 0 0 0 0 0 0 0 ...
 $ cat76.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat77.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat78.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat79.D     : num  0 0 0 0 1 1 0 1 1 0 ...
 $ cat79.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat80.D     : num  1 1 0 1 0 0 1 0 0 0 ...
 $ cat80.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat81.D     : num  1 1 1 1 1 1 1 1 0 0 ...
 $ cat81.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat82.B     : num  1 0 1 0 1 1 1 0 1 1 ...
 $ cat82.OTHER : num  0 0 0 1 0 0 0 0 0 0 ...
 $ cat83.B     : num  0 1 0 1 1 1 0 1 1 1 ...
 $ cat83.OTHER : num  1 0 1 0 0 0 1 0 0 0 ...
 $ cat84.C     : num  1 1 1 1 1 1 1 1 1 1 ...
 $ cat84.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat85.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat86.D     : num  1 1 0 1 0 0 0 1 1 1 ...
 $ cat86.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat87.D     : num  0 0 0 0 0 0 0 0 1 0 ...
 $ cat87.OTHER : num  0 0 0 0 1 0 0 1 0 1 …
 $ cat88.D     : num  0 0 0 0 0 0 0 1 0 1 ...
 $ cat88.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat89.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat90.B     : num  0 0 0 0 0 0 0 0 1 0 ...
 $ cat90.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat91.B     : num  0 0 0 0 1 0 0 0 0 1 ...
 $ cat91.G     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat91.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cat92.H     : num  0 0 0 0 1 0 0 0 0 1 ...
```

Preprocessing the data

```
$ cat92.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat93.D     : num  1 1 1 1 1 1 1 0 1 1 ...
$ cat93.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat94.C     : num  0 0 0 0 0 0 0 0 1 0 ...
$ cat94.D     : num  0 1 1 1 0 1 1 0 0 0 ...
$ cat94.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat95.D     : num  0 0 0 0 1 1 1 0 0 0 ...
$ cat95.E     : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat95.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat96.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat97.C     : num  0 0 0 0 0 1 1 0 1 0 ...
$ cat97.E     : num  0 1 1 1 1 0 0 0 0 0 ...
$ cat97.G     : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat97.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat98.C     : num  1 0 0 0 0 0 0 1 0 1 ...
$ cat98.D     : num  0 1 0 1 0 0 0 0 1 0 ...
$ cat98.OTHER : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat99.P     : num  0 0 0 0 1 1 1 0 0 0 ...
$ cat99.R     : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat99.T     : num  1 1 0 1 0 0 0 1 1 1 ...
$ cat100.F    : num  0 0 0 0 1 0 0 0 0 1 ...
$ cat100.G    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat100.H    : num  0 0 0 0 0 0 0 1 0 0 ...
$ cat100.I    : num  0 0 0 1 0 0 0 0 0 0 ...
$ cat100.J    : num  0 0 0 0 0 1 1 0 0 0 ...
$ cat100.K    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat100.L    : num  0 1 1 0 0 0 0 0 0 0 ...
$ cat100.OTHER: num  1 0 0 0 0 0 0 0 1 0 ...
$ cat101.C    : num  0 0 0 0 0 0 0 1 0 0 ...
$ cat101.D    : num  0 0 0 1 0 1 0 0 0 1 ...
$ cat101.F    : num  0 1 0 0 0 0 0 0 0 0 ...
$ cat101.G    : num  1 0 0 0 0 0 0 0 0 0 ...
$ cat101.OTHER: num  0 0 1 0 1 0 0 0 1 0 ...
$ cat102.OTHER: num  0 0 0 0 0 0 0 0 0 0 ...
$ cat103.B    : num  0 0 1 0 0 0 0 0 0 0 ...
$ cat103.C    : num  0 0 0 0 0 0 1 0 1 0 ...
$ cat103.OTHER: num  0 0 0 0 0 0 0 0 0 1 ...
$ cat104.E    : num  0 1 1 1 0 1 1 0 0 0 ...
```

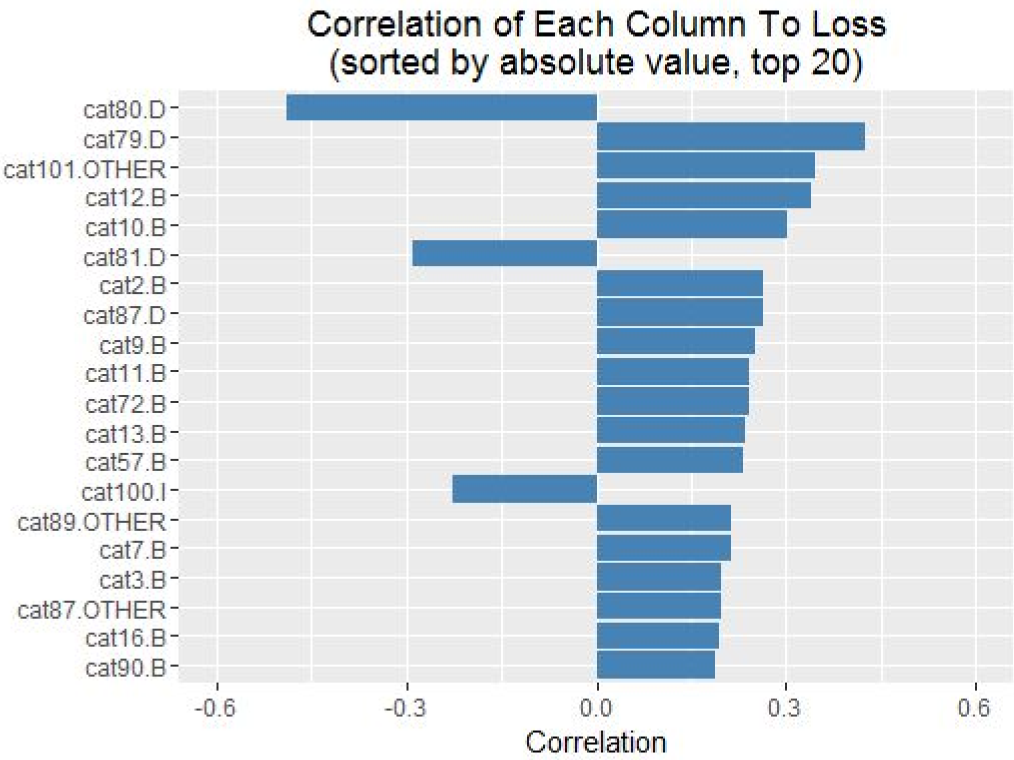
```
$ cat104.F    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat104.G    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat104.H    : num  0 0 0 0 0 0 0 0 1 0 ...
$ cat104.I    : num  1 0 0 0 0 0 0 0 0 0 ...
$ cat104.K    : num  0 0 0 0 0 0 0 1 0 1 ...
$ cat104.OTHER: num  0 0 0 0 0 0 0 0 0 0 ...
$ cat105.E    : num  1 1 0 1 1 1 1 0 0 0 ...
$ cat105.F    : num  0 0 1 0 0 0 0 1 1 0 ...
$ cat105.G    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat105.H    : num  0 0 0 0 0 0 0 0 0 1 ...
$ cat105.OTHER: num  0 0 0 0 0 0 0 0 0 0 ...
$ cat106.F    : num  0 0 0 0 0 0 0 1 0 0 ...
$ cat106.G    : num  1 0 0 0 0 0 0 0 1 1 ...
$ cat106.H    : num  0 0 1 0 0 1 1 0 0 0 ...
$ cat106.I    : num  0 1 0 1 0 0 0 0 0 0 ...
$ cat106.J    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat106.OTHER: num  0 0 0 0 1 0 0 0 0 0 ...
$ cat107.F    : num  0 0 1 0 0 1 1 0 0 0 ...
$ cat107.G    : num  0 0 0 0 1 0 0 0 0 0 ...
$ cat107.H    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat107.I    : num  0 0 0 0 0 0 0 1 0 0 ...
$ cat107.J    : num  1 0 0 0 0 0 0 0 0 1 ...
$ cat107.K    : num  0 1 0 1 0 0 0 0 0 0 ...
$ cat107.OTHER: num  0 0 0 0 0 0 0 0 1 0 ...
$ cat108.D    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat108.F    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat108.G    : num  1 0 0 0 0 0 0 1 0 1 ...
$ cat108.K    : num  0 1 0 1 0 0 0 0 1 0 ...
$ cat108.OTHER: num  0 0 1 0 0 0 0 0 0 0 ...
$ cat109.BI   : num  0 1 0 1 0 1 1 1 1 0 ...
$ cat109.OTHER: num  1 0 0 0 1 0 0 0 0 1 ...
$ cat110.CL   : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat110.CO   : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat110.CS   : num  0 0 0 1 0 1 0 0 0 0 ...
$ cat110.EB   : num  0 0 0 0 0 0 0 1 0 0 ...
$ cat110.EG   : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat110.OTHER: num  1 1 1 0 1 0 1 0 1 1 ...
$ cat111.C    : num  1 0 0 1 1 0 0 0 1 0 ...
```

```
$ cat111.E    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat111.OTHER: num  0 0 0 0 0 0 0 1 0 0 ...
$ cat112.AS   : num  1 0 0 0 0 1 0 0 0 0 ...
$ cat112.E    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat112.J    : num  0 0 0 0 0 0 1 0 0 0 ...
$ cat112.OTHER: num  0 1 1 1 1 0 0 0 1 1 ...
$ cat113.AX   : num  0 0 0 0 0 0 0 0 1 0 ...
$ cat113.BM   : num  0 1 0 0 1 0 0 0 0 0 ...
$ cat113.L    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat113.OTHER: num  1 0 1 0 0 0 1 0 0 1 ...
$ cat113.Y    : num  0 0 0 0 0 0 0 1 0 0 ...
$ cat114.C    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat114.E    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat114.OTHER: num  0 0 0 0 0 0 0 0 0 1 ...
$ cat115.K    : num  0 0 0 0 1 1 1 0 0 0 ...
$ cat115.L    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat115.M    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat115.N    : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat115.O    : num  1 1 0 1 0 0 0 0 0 1 ...
$ cat115.OTHER: num  0 0 1 0 0 0 0 0 1 0 ...
$ cat115.P    : num  0 0 0 0 0 0 0 1 0 0 ...
$ cat116.DJ   : num  0 0 0 1 0 1 1 0 0 0 ...
$ cat116.HK   : num  0 0 0 0 0 0 0 0 0 0 ...
$ cat116.OTHER: num  1 1 1 0 0 0 0 1 1 1 ...
$ cont1       : num  0.726 0.331 0.262 0.322 0.273 ...
$ cont2       : num  0.246 0.737 0.358 0.556 0.16 ...
$ cont3       : num  0.188 0.593 0.484 0.528 0.528 ...
$ cont4       : num  0.79 0.614 0.237 0.374 0.473 ...
$ cont5       : num  0.31 0.886 0.397 0.422 0.704 ...
$ cont6       : num  0.718 0.439 0.29 0.441 0.178 ...
$ cont7       : num  0.335 0.437 0.316 0.391 0.247 ...
$ cont8       : num  0.303 0.601 0.273 0.318 0.246 ...
$ cont9       : num  0.671 0.351 0.261 0.321 0.221 ...
$ cont10      : num  0.835 0.439 0.324 0.445 0.212 ...
$ cont11      : num  0.57 0.338 0.381 0.328 0.205 ...
$ cont12      : num  0.595 0.366 0.373 0.322 0.202 ...
$ cont13      : num  0.822 0.611 0.196 0.605 0.246 ...
$ cont14      : num  0.715 0.304 0.774 0.603 0.433 ...
$ loss        : num  7.79 7.3 8.07 7.04 7.99 ...
```

Variable Correlation To Loss

	variable	cor
80	cat80.D	-0.4881326832
78	cat79.D	0.4260899524
136	cat101.OTHER	0.3453149606
11	cat12.B	0.3412501993
9	cat10.B	0.3028827760
82	cat81.D	-0.2913617979
1	cat2.B	0.2648029266
93	cat87.D	0.2642634448
8	cat9.B	0.2518229180
10	cat11.B	0.2423092410
71	cat72.B	0.2420830526
12	cat13.B	0.2362243575
56	cat57.B	0.2306959711
127	cat100.I	-0.2276397230
6	cat7.B	0.2136900721
97	cat89.OTHER	0.2136900721
2	cat3.B	0.1983373842
94	cat87.OTHER	0.1965621859
15	cat16.B	0.1956941440
98	cat90.B	0.1867979769
22	cat23.B	0.1820766881
72	cat73.OTHER	-0.1818555804
35	cat36.B	0.1771345930
5	cat6.B	-0.1654783556
125	cat100.G	0.1612349228
140	cat103.OTHER	0.1598830127
49	cat50.B	-0.1597130689
181	cat111.OTHER	0.1528579695
191	cat114.C	-0.1500996867

	variable	cor
130	cat100.L	0.1481612062
192	cat114.E	-0.1481490690
102	cat91.OTHER	0.1335827252
27	cat28.B	0.1316297212
39	cat40.B	0.1300847358
4	cat5.B	0.1296647006
3	cat4.B	0.1241861659
37	cat38.B	0.1233270431
83	cat81.OTHER	0.1198687729
128	cat100.J	-0.1155922480
85	cat82.OTHER	-0.1141594350
24	cat25.B	0.1131868534
205	cont2	0.1086686454
75	cat76.OTHER	0.1021539383
23	cat24.B	0.1010686024
40	cat41.B	0.0959454097
7	cat8.B	0.0949661670
137	cat102.OTHER	0.0949661670
13	cat14.B	0.0941906280
126	cat100.H	0.0909553429
210	cont7	0.0869832821
206	cont3	0.0846076161
28	cat29.B	0.0838642608
131	cat100.OTHER	0.0828205770
44	cat45.B	0.0801435868
84	cat82.B	0.0794583657
43	cat44.B	0.0793613996
214	cont11	0.0740673047
215	cont12	0.0735169506
...		



Linear Model

```
library(caret)
set.seed(0)
inTrain1<- createDataPartition(y=new.all.cd.train$loss, p=0.80, list=FALSE, times=1)
training<-new.all.cd.train[inTrain1,]
testing<-new.all.cd.train[-inTrain1,]

lmFit1 <- train(loss~., data=training, method='lm')

lmFit1adj2 <- train(loss~. - cat114.OTHER -cat111.OTHER -cat103.OTHER -cat101.OTHER
                    -cat102.OTHER -cat90.OTHER -cat89.OTHER, data=training, method='lm')

lmFit1adj3 <- train(loss~. - cat114.OTHER -cat111.OTHER -cat103.OTHER -cat101.OTHER
                    -cat102.OTHER -cat90.OTHER -cat89.OTHER -cat6.B -cat8.B -cat10.B
                    -cat10.B -cat15.B -cat19.B -cat19.B -cat24.B -cat30.B -cat33.B
                    -cat43.B -cat45.B -cat46.B -cat58.B -cat60.B -cat62.B -cat64.B
                    -cat66.B -cat68.B -cat69.B -cat70.B -cat81.OTHER -cat82.B -cat82.B
                    -cat83.B -cat84.OTHER -cat86.D -cat88.D -cat88.OTHER -cat92.OTHER
                    -cat96.OTHER -cat97.C -cat97.E -cat97.OTHER -cat98.C -cat98.D
                    -cat98.OTHER -cat99.R -cat99.T -cat100.I -cat104.F -cat104.G -cat104.H
                    -cat104.K -cat104.OTHER -cat105.E -cat105.F -cat105.H -cat106.F
                    -cat106.G -cat106.J -cat107.H -cat108.F -cat108.G -cat108.G -cat109.BI
                    -cat109.OTHER -cat110.CL -cat110.CO -cat110.EG -cat110.OTHER -cat113.AX
                    -cat113.OTHER -cat115.K -cat115.L -cat115.L -cat115.M -cat115.N -cat115.N
                    -cat115.O -cat115.OTHER -cat115.P -cont3 -cont5 -cont6 -cont13,
                    data=training, method='lm')
```

summary(lmFit1)

Includes all variables

```
Residual standard error: 0.5067 on 150443 degrees of freedom
Multiple R-squared:  0.5215,     Adjusted R-squared:  0.5208
F-statistic: 773.3 on 212 and 150443 DF,  p-value: < 2.2e-16
```

summary(lmFit1adj2)

Excludes all variables with NA coefficients in lmFit1

```
Residual standard error: 0.5067 on 150443 degrees of freedom
Multiple R-squared:  0.5215,     Adjusted R-squared:  0.5208
F-statistic: 773.3 on 212 and 150443 DF,  p-value: < 2.2e-16
```

summary(lmFit1adj3)

Excludes all variables not significant at least at the 90% confidence level in lmFit1

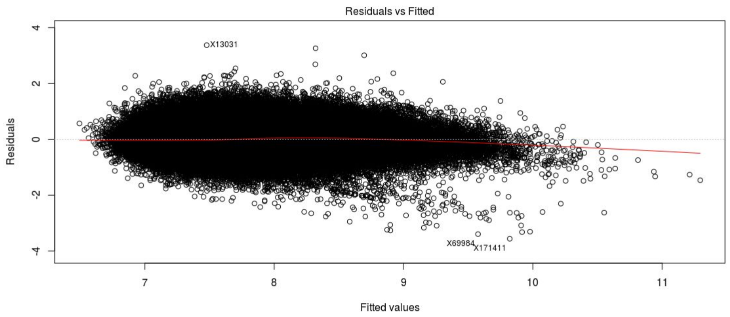
```
Residual standard error: 0.507 on 150513 degrees of freedom
Multiple R-squared:  0.5208,     Adjusted R-squared:  0.5203
F-statistic:  1152 on 142 and 150513 DF,  p-value: < 2.2e-16
```

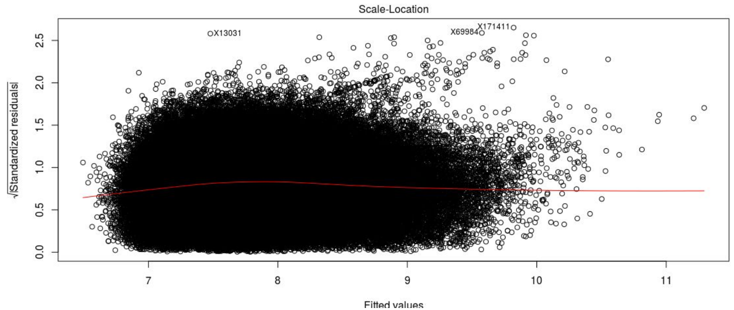
Linear Model

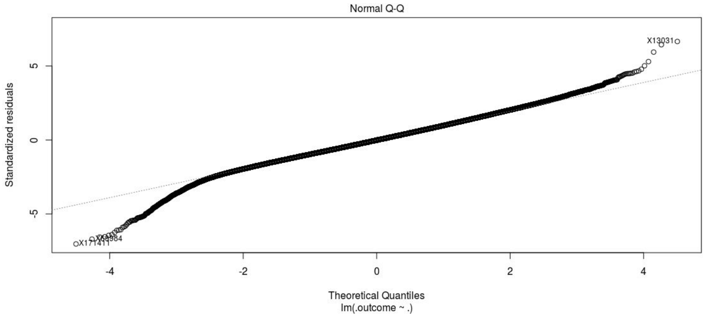
```
varImp(lmFit1adj3, scale = FALSE)

lm variable importance

only 20 most important variables
shown (out of 142)

              Overall
cat80.D         71.46
cat53.B         51.28
cat79.D         49.50
cat81.D         40.12
cat100.G        39.95
cat100.L        38.19
cat112.J        37.95
cat2.B          32.56
cat100.H        31.89
cat101.C        29.60
cat112.OTHER    29.50
cat101.D        29.23
cat72.B         28.23
cat26.B         27.96
cat44.B         27.91
cont2           27.45
cat12.B         26.34
cat1.B          25.88
cat100.OTHER    23.22
cont7           22.37
```

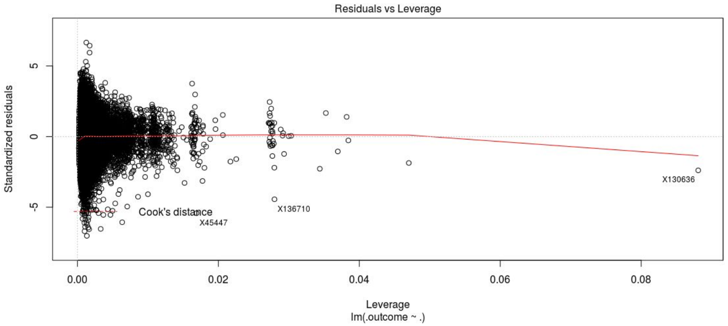
RMSE:
0.5054915

MAE for the model (not Kaggle score):
0.397099







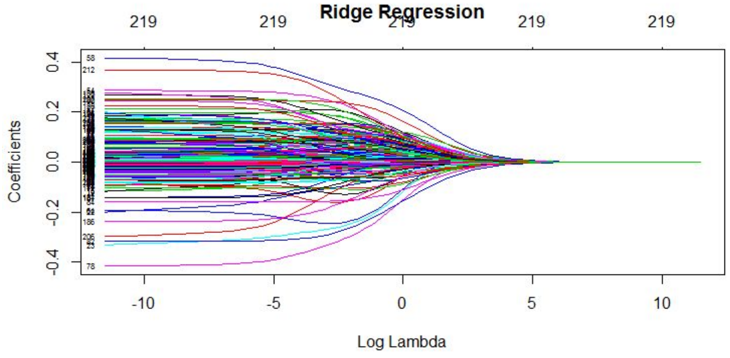


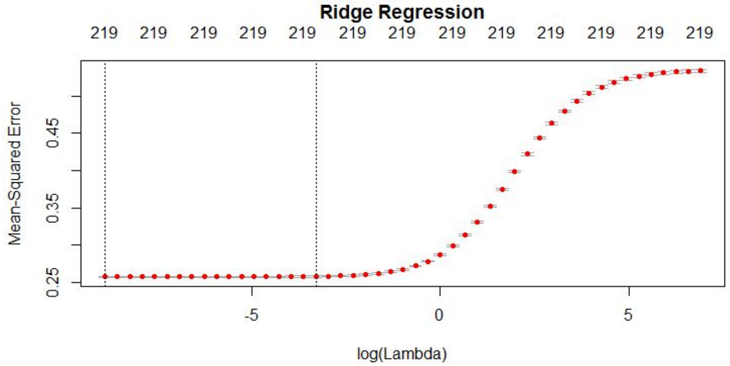
Ridge Model

lambda	RMSE	Rsquared
1.000000e-05	0.5071109	0.5200358
2.212216e-05	0.5075730	0.5192098
3.290345e-05	0.5069499	0.5199159
4.893901e-05	0.5072538	0.5197626
7.278954e-05	0.5072210	0.5196924
1.082637e-04	0.5071089	0.5200397
2.395027e-04	0.5067146	0.5208755
3.562248e-04	0.5075697	0.5192165
5.298317e-04	0.5069426	0.5199301
7.880463e-04	0.5072481	0.5197738
1.172102e-03	0.5071043	0.5200493
2.592944e-03	0.5070029	0.5206113
3.856620e-03	0.5067233	0.5208621
5.736153e-03	0.5075894	0.5191892
8.531679e-03	0.5069802	0.5198728
1.268961e-02	0.5071969	0.5199062
2.807216e-02	0.5075871	0.5191559
4.175319e-02	0.5076327	0.5197923
6.210169e-02	0.5078219	0.5195180
9.236709e-02	0.5096351	0.5170850
1.373824e-01	0.5108307	0.5168964
3.039195e-01	0.5205381	0.5131789
4.520354e-01	0.5334754	0.5099944
6.723358e-01	0.5577743	0.5070403
1.000000e+00	0.6006881	0.5015040

RMSE was used to select the optimal model using the smallest value.

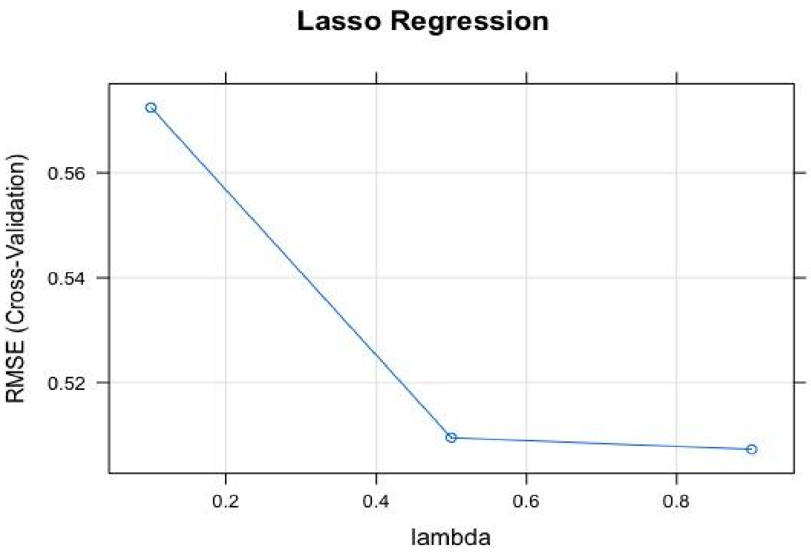
The final value used for the model was lambda = 0.0002395027.
Kaggle score of 1232
RMSE: 0.5067146





Lasso model

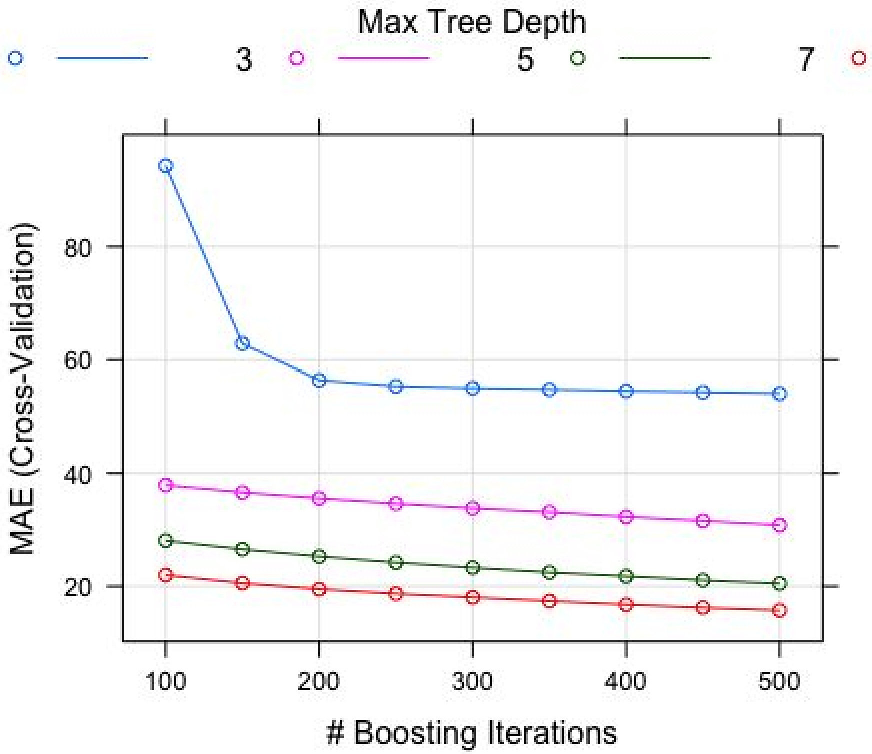
```
150656 samples
   218 predictor

Pre-processing: scaled (218), centered (218)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 135590, 135590, 135591, 135591,
Resampling results across tuning parameters:

  fraction  RMSE       Rsquared
  0.1       0.5724427  0.4095845
  0.5       0.5094885  0.5153269
  0.9       0.5072764  0.5189440

RMSE was used to select the optimal model using  the
 smallest value.
The final value used for the model was fraction = 0.9.
```



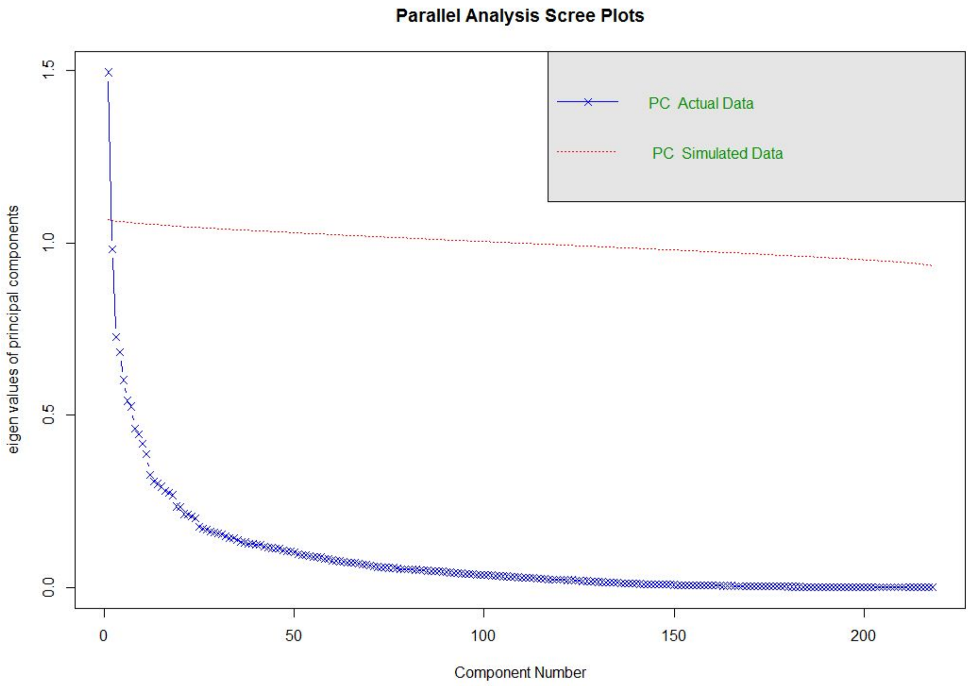
RMSE for the model: 0.505312
MAE for the model (not Kaggle score) 0.3978363

GBM

- GBM was done on the pre-processed dataset.
- The following parameters were used:
  - N.trees - 500
  - Interaction depth - 1,3,5,7
  - Shrinkage - 0.1
- An MAE of 1245.942 was achieved on a subset of the train dataset.



PCA

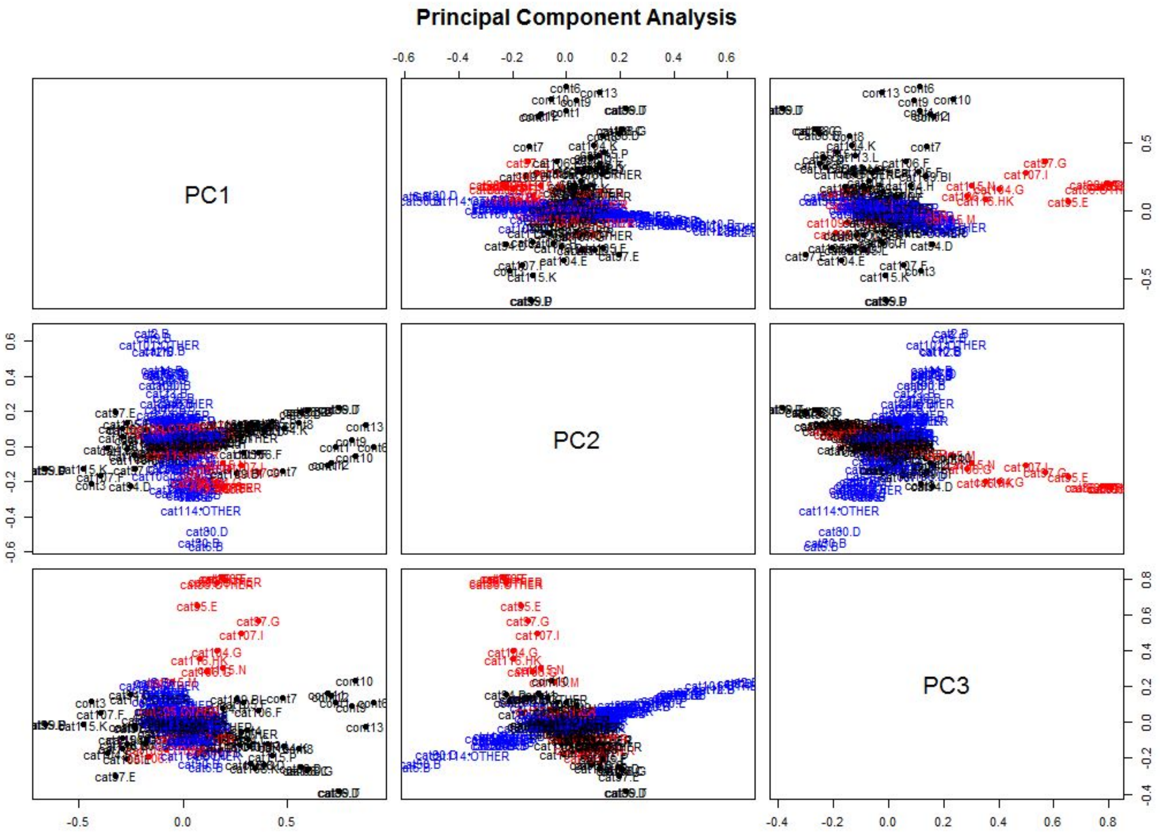


Parallel Analysis Scree Plots
PC Actual Data
PC Simulated Data
eigen values of principal components
Component Number
0.0
0.5
1.0
1.5
0
50
100
150
200

PCA

	PC1	PC2	PC3
SS loadings	12.37	7.03	6.38
Proportion Var	0.06	0.03	0.03
Cumulative Var	0.06	0.09	0.12
Proportion Explained	0.48	0.27	0.25
Cumulative Proportion	0.48	0.75	1.00

```
Mean item complexity =  1.6
Test of the hypothesis that 3 components are
sufficient.

The root mean square of the residuals (RMSR)
is  0.06

Fit based upon off diagonal values = 0.53
```

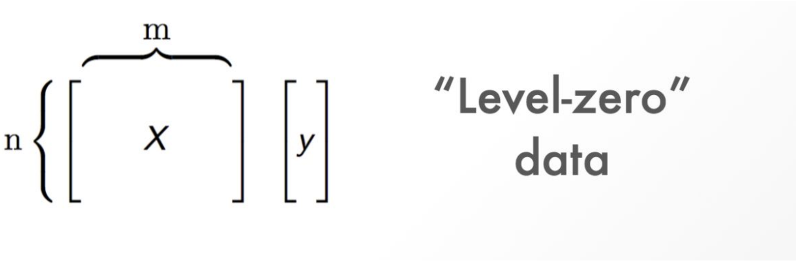


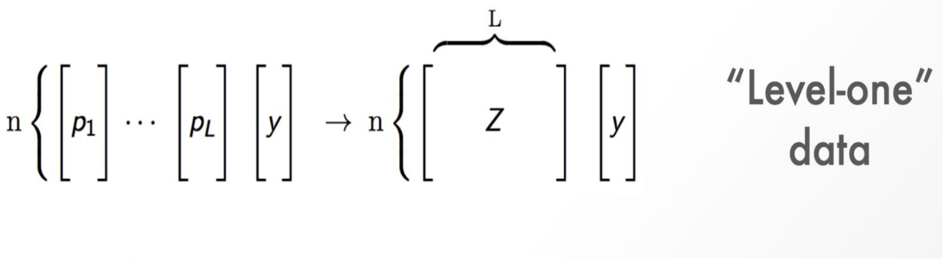
Ensembling

- None of the models did well on its own.
- But choosing from the models and parameters we tried, we assembled a group of learners.
- H2O and H2O ensemble was used.

Ensembling

- Start with L base learners (each with its own model parameters)
  - Base learners will be trained on the “Level-zero data” to produce L number of predictions, p.
- Column bind all predictions, p.
  - These will be the new predictors for response, y.
- Specify a metalearner.
  - Metalearner will be used on the “Level-one data”



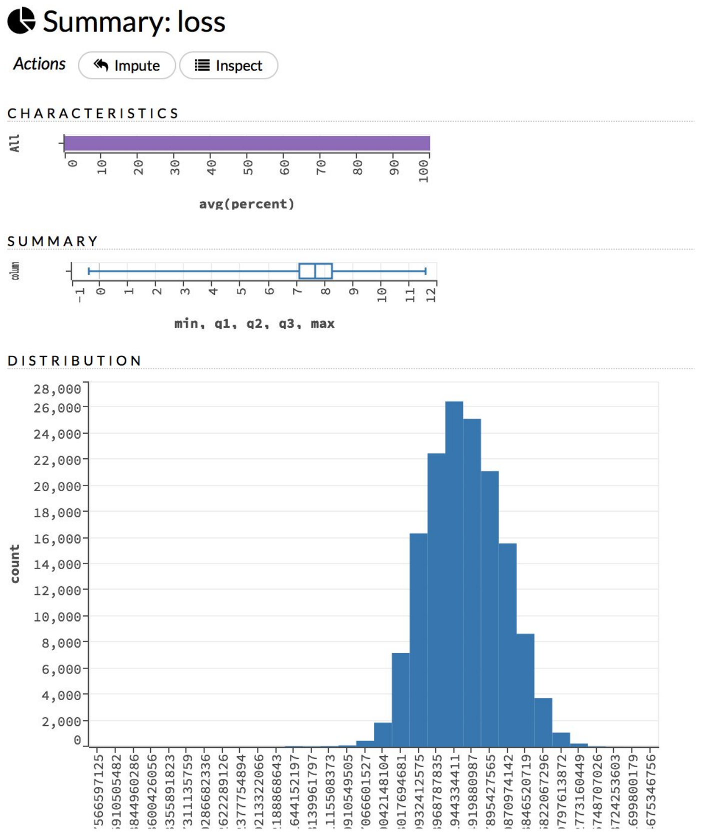


Ensembling

- Creating learners with parameters:
  - h2o.glm.3 <- function(..., alpha = 1.0) h2o.glm.wrapper(..., alpha = alpha)
  - h2o.randomForest.1 <- function(..., ntrees = 300)
  - h2o.gbm.3 <- function(..., ntrees = 500, max_depth = 7, seed = 1)
  - h2o.deeplearning.1 <- function(..., hidden = c(500,500), epochs = 50, seed = 1)
- Setup base learners and metalearner to be used on Level-zero data:
  - learner <- c("h2o.glm.wrapper", "h2o.randomForest.1", "h2o.gbm.3", "h2o.deeplearning.wrapper", “h2o.deeplearning.1”))
  - metalearner <- "h2o.gbm.1"
- Train & test:
  - fit <- h2o.ensemble(x = x, y = y, data = train, family = family, learner = learner, metalearner = metalearner)
  - pred <- predict(fit = fit, newdata = test)

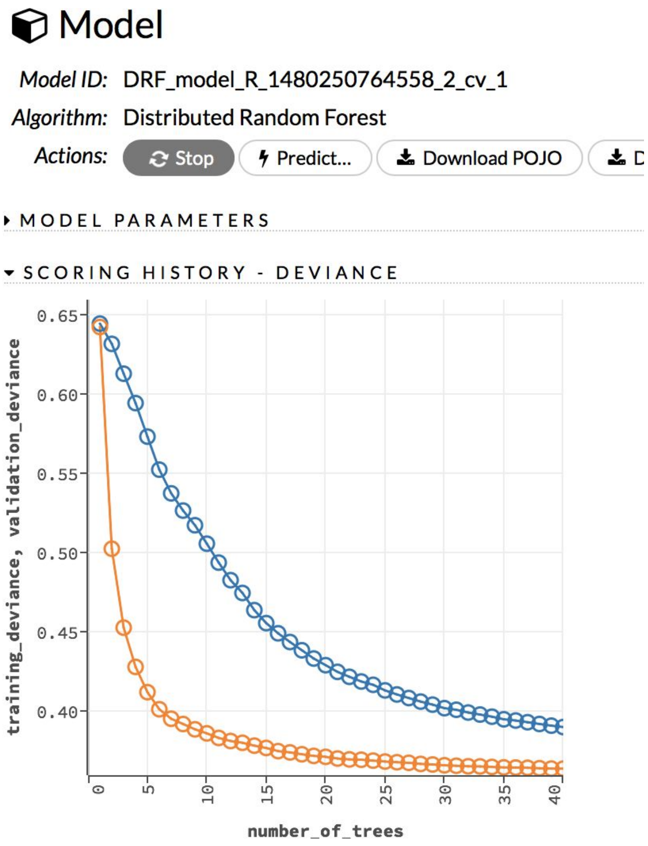
Ensembling

- H2O runs outside of R. JRE must be installed on the machine.
- Has a separate web interface to show what's going on:



Ensembling

- Can show model performance real-time:



Ensembling

- Shows Job queue and REMAINING TIME!!!

Job

Run Time	00:10:30.786
Remaining Time	07:20:02.900
Type	Model
Key	DRF_model_R_1480250764558_2
Description	DRF
Status	RUNNING
Progress	3%
	Scoring the model.
Actions	Cancel Job

Ensembling

- Base learners:
  - Glm:
    - Lambda = 1e-5
  - RandomForest:
    - N.trees = 300
    - Max_depth = 20
  - Gbm.1:
    - N.trees = 500
    - Max_depth = 5
  - Gbm.2:
    - N.trees = 300
    - Max_depth = 5
  - Gbm.3:
    - N.trees = 300
    - Max_depth = 3
  - Deeplearning
    - Hidden = c(20,20)
    - Epochs = 10
- Metalearner:
  - Gbm.1
- Kaggle score of 1125.39604

Thank you!