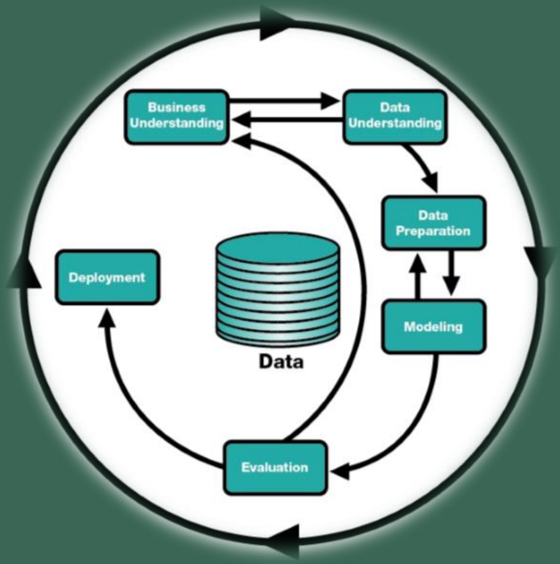
Kaggle Fire Loss Prediction

A Strategic Approach with CRISP-DM Model

Presented by FireKeeper

Zhongxing Xue, Yuqi Zhang, Zexing Xu, Liangchen Xing, Tonghong Chen

Nov. 5, 2017



- BUSINESS UNDERSTANDING
- DATA UNDERSTANDING
- DATA PREPARATION
- MODELING
- EVALUATION
- CONCLUSION (DEPLOYMENT)

Our Approach

Cross-Industry standard process for data mining (CRISP-DM)

BUSINESS UNDERSTANDING

Situation

Liberty Mutual Insurance wants to better estimate its policyholder's risk to increase its profit

Objective

Predict the Risk of a property and help the company to increase profit

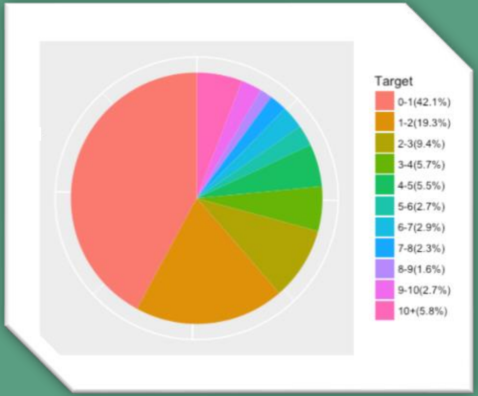
DATA UNDERSTANDING

Pros

- ✓ Most features have decent amount of data
- ✓ Most features have been normalized, allowing us to fill missing data with 0 (average)

Cons

- ✓ Contains categorical variables -> Need to be transformed into numeric variables
- ✓ features such as weather115 have only one value
- ✓ *Artificial features such as id have no contribution to the prediction*



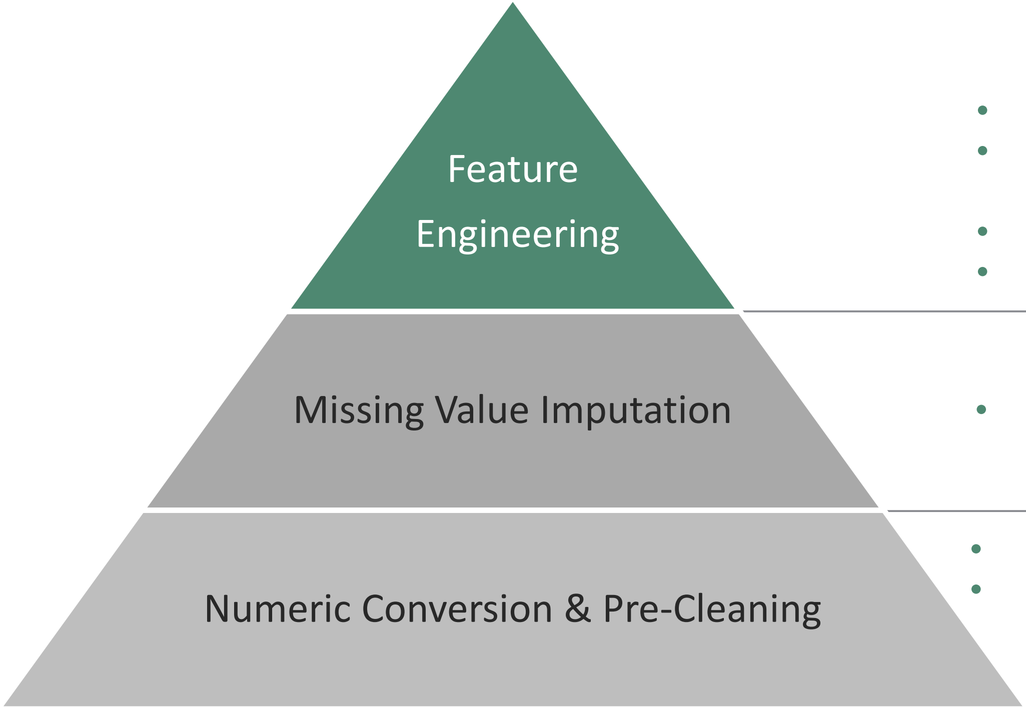
Distribution of Target (>0) in training dataset

eatherVar207	weatherVar208	weatherVar209	weatherVar210	weatherVar211
19386	19386	32373	32373	32373
weatherVar212	weatherVar213	weatherVar214	weatherVar215	weatherVar216
32373	32373	32373	32373	32373
weatherVar217	weatherVar218	weatherVar219	weatherVar220	weatherVar221
32373	32373	32373	32373	32373
weatherVar222	weatherVar223	weatherVar224	weatherVar225	weatherVar226
32373	32373	32373	32373	32373
var15	crimeVar1	crimeVar3	crimeVar6	crimeVar8
98856	109988	109988	109988	109988
crimeVar9	crimeVar5	crimeVar4	crimeVar2	crimeVar7
109988	110655	112798	114553	117363
var14	var12	var16		
290466	355042	361693		

Top missing-data features

< >

DATA PREPARATION



- Lasso
- Filter Method: Pearson and Spearman Correlation Coefficient
- RFE (Recursive Feature Elimination)
- PCA (Principle Component Analysis)

- Fill all missing data with 0 (mean of the normalized data)

- One-hot Encoding
- Delete *var12, var14, var16, weather115, X, id* from training set

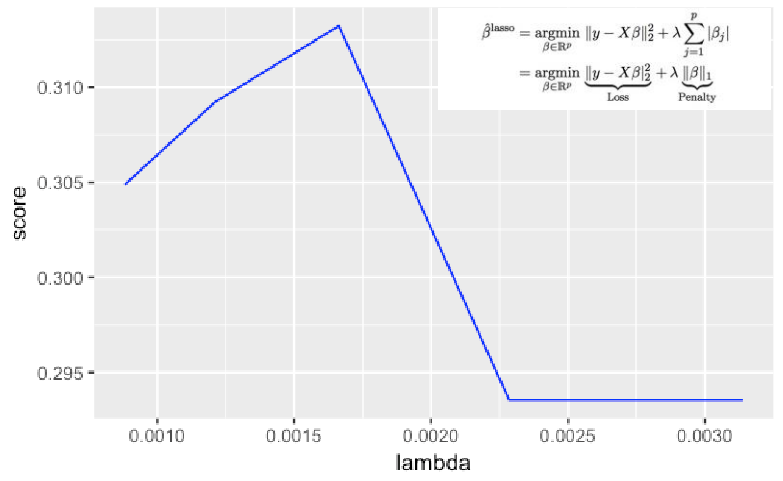
6 MODELING – LASSO REGRESSION

Trained with 287 features. ***Kaggle Leaderboard Rank 11th***

Lambda	3.139e-3	2.285e-3	1.663e-3	1.211e-3	0.881e-3
Kaggle Score	0.29356	0.29356	0.31324	0.30923	0.30488
D.F.	1	1	4	5	10

TOP FOUR VARIABLES

- ✓ var13
- ✓ weatherVar118
- ✓ var15
- ✓ weatherVar102

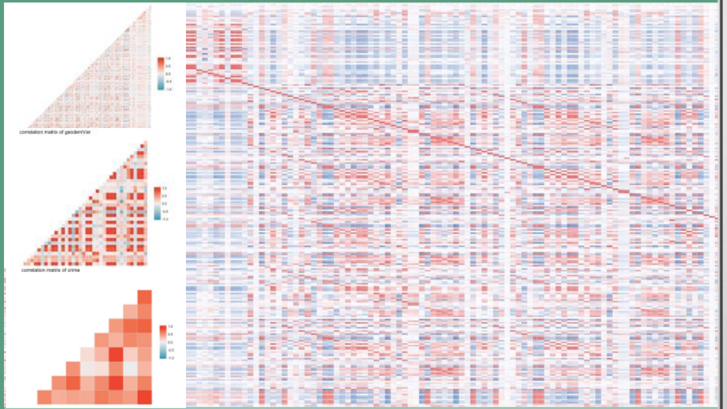


$$\hat{\beta}^{\text{lasso}} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

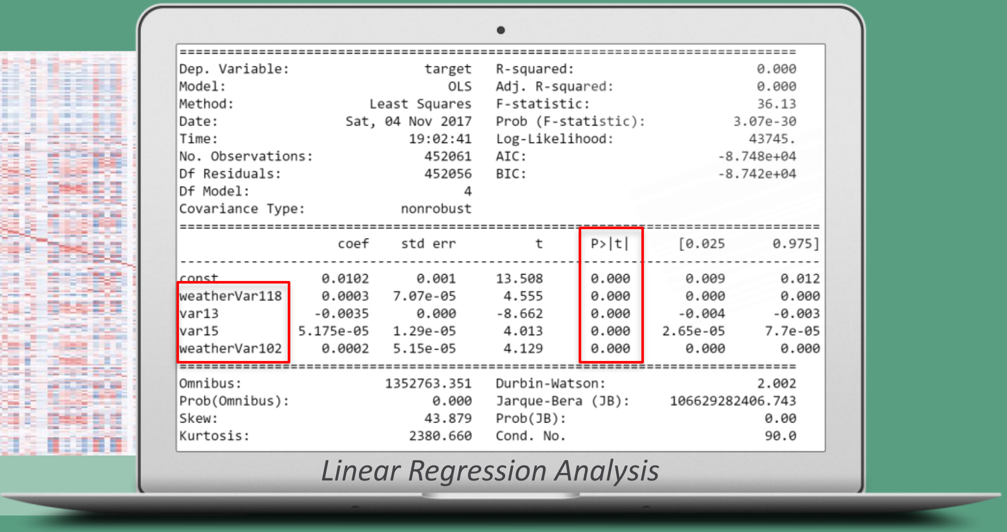
```r
library(glmnet)
y <- as.matrix(Data2[, "target"])
x <- as.matrix(Data2[, c(2 : 287)])
FIT <- glmnet(x, y, family = "gaussian", nlambda = 30,
print(FIT)
coef(FIT, s=FIT$lambda[4])

xNew <- as.matrix(Test)
colnames(xNew)
xNew <- apply(xNew, 2, function(x) Trans_Zero(x))
ID <- xNew[, 1]
xNew <- xNew[, -1]
yNew <- predict(FIT, newx = xNew, s=FIT$lambda[4])
Ans <- matrix(0, nrow(yNew), 2)
Ans <- data.frame(Ans)
Ans[, 1] <- ID
Ans[, 2] <- yNew
names(Ans) <- c("id","target")
write.csv(Ans, file="Sub024.csv", row.names = FALSE)
```

7 MODELING – UNIVARIATE/MULTI-VAR REGRESSION

Try related model to see if we could get higher score



Pearson Correlation Matrix



```
==============================================================================
Dep. Variable:                 target   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                  0.000
Method:                 Least Squares   F-statistic:                     36.13
Date:                Sat, 04 Nov 2017   Prob (F-statistic):           3.07e-30
Time:                        19:02:41   Log-Likelihood:                 43745.
No. Observations:              452061   AIC:                        -8.748e+04
Df Residuals:                  452056   BIC:                        -8.742e+04
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const            0.0102      0.001     13.508      0.000       0.009       0.012
weatherVar118    0.0003   7.07e-05      4.555      0.000       0.000       0.000
var13           -0.0035      0.000     -8.662      0.000      -0.004      -0.003
var15         5.175e-05   1.29e-05      4.013      0.000    2.65e-05     7.7e-05
weatherVar102    0.0002   5.15e-05      4.129      0.000       0.000       0.000
==============================================================================
Omnibus:                  1352763.351   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):  106629282406.743
Skew:                          43.879   Prob(JB):                         0.00
Kurtosis:                    2380.660   Cond. No.                         90.0
```

Linear Regression Analysis

Interesting Finding: var13 tends to be significant everywhere

8

MODELING – OTHER MODELS

Try different models to see if we could get higher score

Ridge Regression

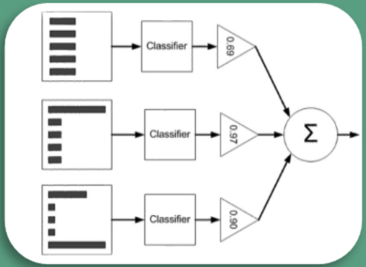
with highest score 0.24

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$$

```r
y <- as.matrix(Data2[, "target"])
x <- as.matrix(Data2[, c(2 : 287)])
FIT <- glmnet(x, y, family = "gaussian", nlambda =
print(FIT)
coef(FIT, s=FIT$lambda[4])

xNew <- as.matrix(Test)
colnames(xNew)
xNew <- apply(xNew, 2, function(x) Trans_Zero(x))
ID <- xNew[, 1]
xNew <- xNew[, -1]
yNew <- predict(FIT, newx = xNew, s=FIT$lambda[4])
Ans <- matrix(0, nrow(yNew), 2)
Ans <- data.frame(Ans)
Ans[, 1] <- ID
```

AdaBoosting with Decision Tree Regressor

with highest score 0.03

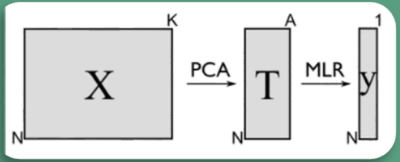


ADA BOOST

```python
train_ada = train.drop(['var12', 'var14', 'var16', 'weatherVar115', 'id'], 1)
fill_dataset(train_ada)

test_ada = test.drop(['var12', 'var14', 'var16', 'weatherVar115', 'id'], 1)
fill_dataset(test_ada)

adaboost = sklearn.ensemble.AdaBoostRegressor(n_estimators=1000)
adaboost.fit(train_ada.drop('target',1), train_ada['target'])

AdaBoostRegressor(base_estimator=None, learning_rate=1.0, loss='linear',
         n_estimators=1000, random_state=None)

result = adaboost.predict(test_ada)
```

Principle Component Regression

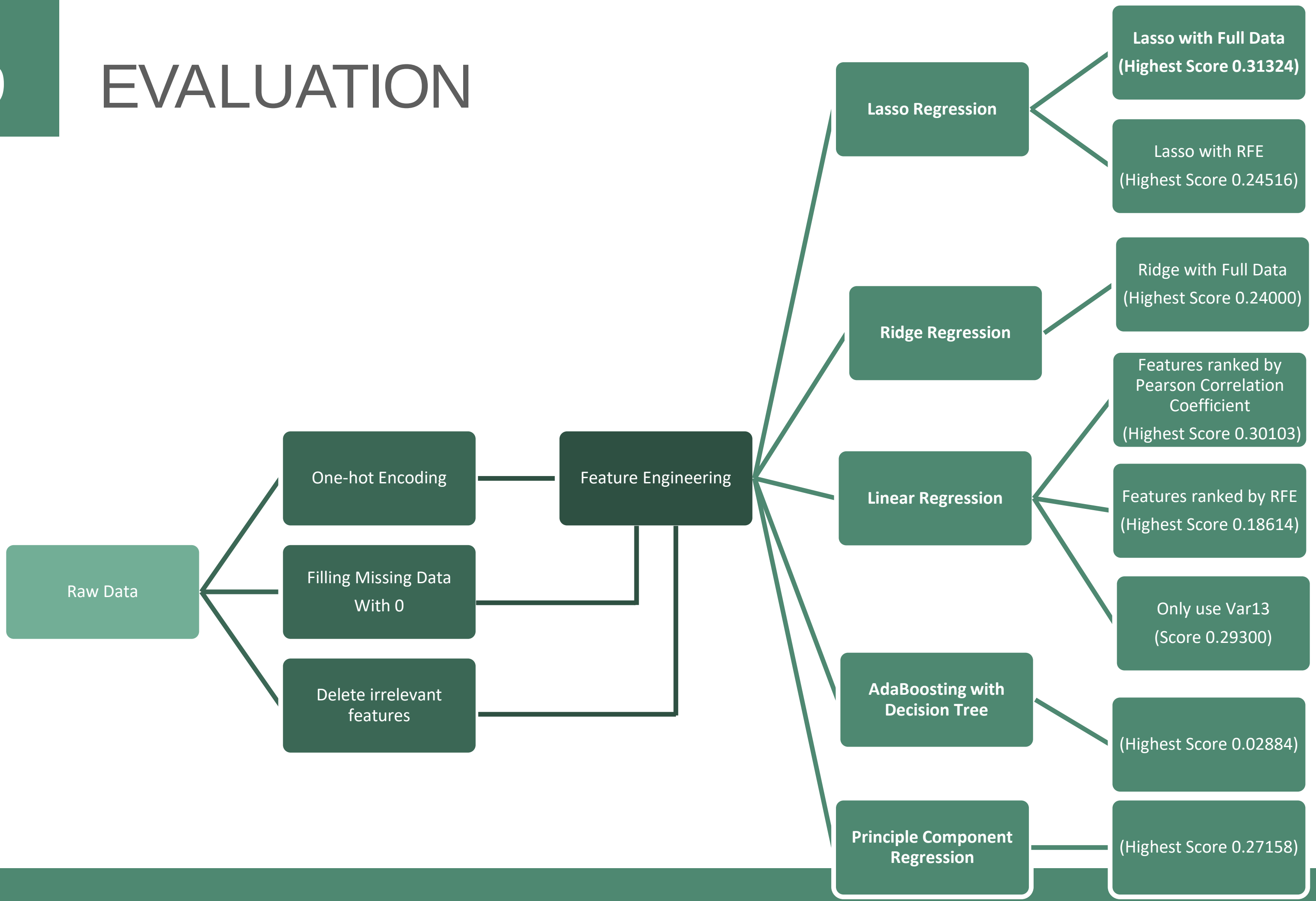
with highest score 0.27



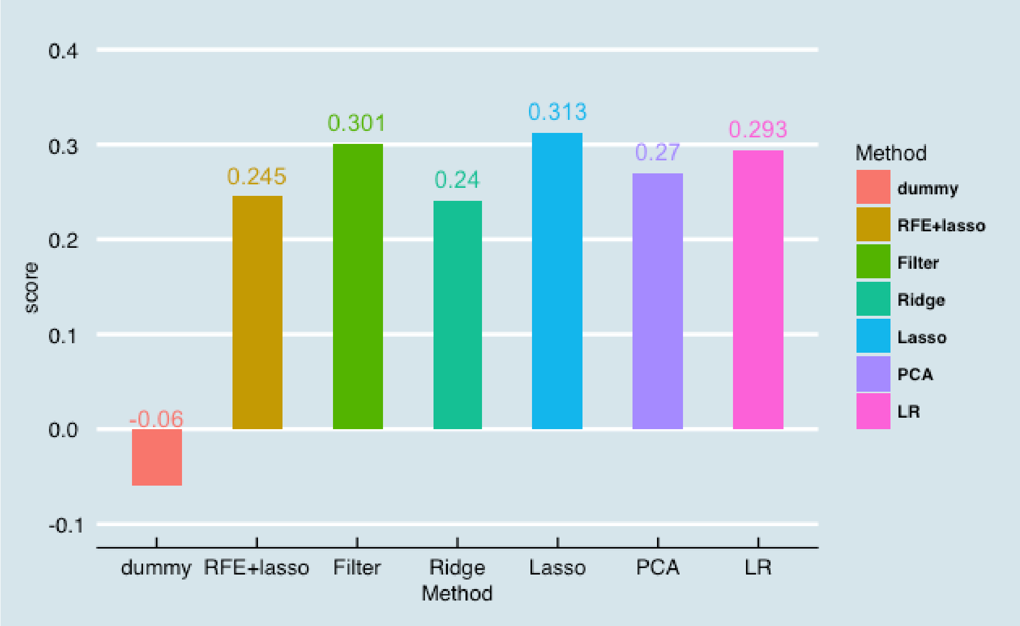

```r
library(factoextra)
pca1 <- princomp(sub_x[,c(5:160,161:169,171:291)]) # doin

model1 <- glm(sub_x$target ~ pca1$scores[, 1:11])
summary(model1)
class(pca1$scores)
pca1.score <- data.frame(pc1=pca1$scores[, 1],
                         pc2=pca1$scores[, 2],
                         pc3=pca1$scores[, 3],
                         pc4=pca1$scores[, 4],
                         pc5=pca1$scores[, 5],
                         pc6=pca1$scores[, 6],
                         pc7=pca1$scores[, 7],
                         pc8=pca1$scores[, 8],
                         pc9=pca1$scores[, 9],
                         pc10=pca1$scores[, 10],
                         pc11=pca1$scores[, 11])
pred_target1 <- predict.glm(model1, newdata = pca1.score)
```

< >

9 EVALUATION



EVALUATION



Pros

- ✓ Consider all the variables and their correlation.
- ✓ Transfer categorical variables into numeric ones
- ✓ Fill in the blanks for missing data
- ✓ Reach a condensed conclusion from different models

Next Steps

- ✓ Try different boosting methods to solve the problem that the distribution of data is unbalanced.
- ✓ Based on the characteristics of categorical variables, we could try more combinations of classification and regression model.

CONCLUSION

Keys to address insurance problem

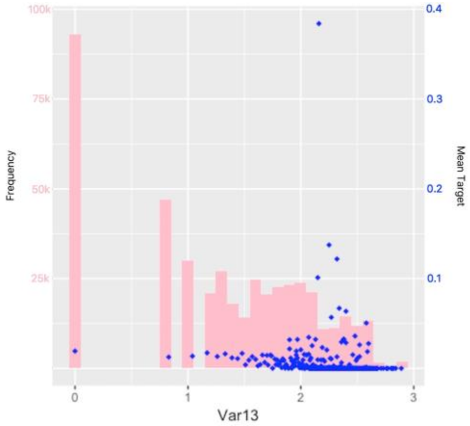
- ✓ Feature Reduction
- ✓ Missing Data Imputation
- ✓ Simplicity > Complexity

Magic of Var13

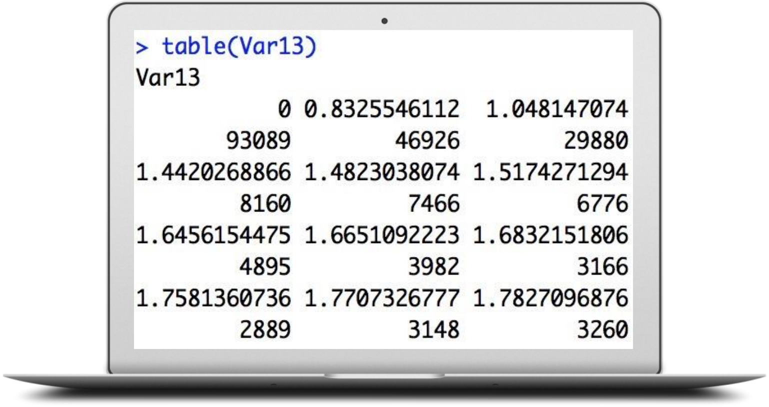
- ✓ Key feature for fire loss prediction

Profit Maximization

- ✓ Use our Lasso Model to ace the prediction
- ✓ Pay attention to the tail that cause a great loss (high value in *target*)



Var13 = sqrt(ln(N))



```
> table(Var13)
Var13
           0 0.8325546112  1.048147074
       93089        46926        29880
1.4420268866 1.4823038074 1.5174271294
        8160         7466         6776
1.6456154475 1.6651092223 1.6832151806
        4895         3982         3166
1.7581360736 1.7707326777 1.7827096876
        2889         3148         3260
```

THANKS!