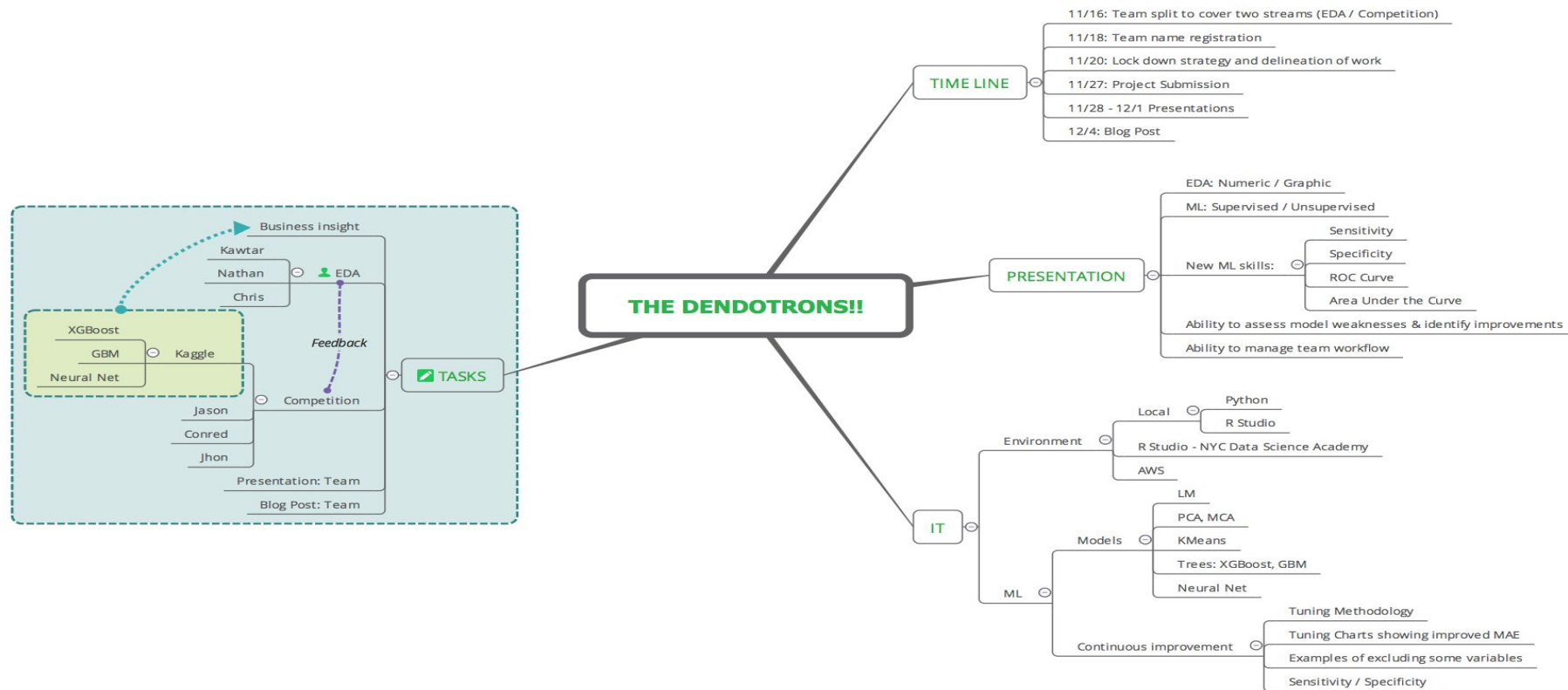




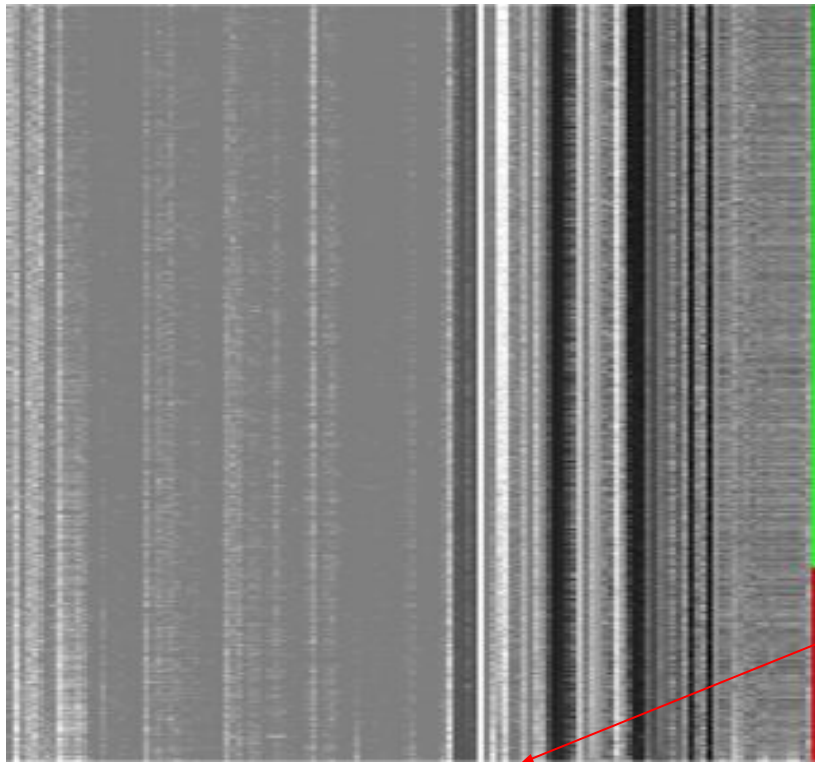
# The Dendotrons Kaggle: Allstate

Kawtar Belmkaddem  
Jhonasttan Regalado  
Jason Sippie  
Nathan Stevens  
Chris Valle  
Conred Wang

# Team Development / Management

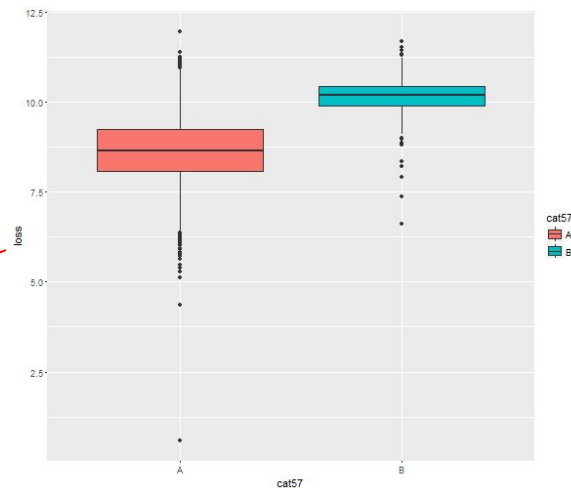


# EDA



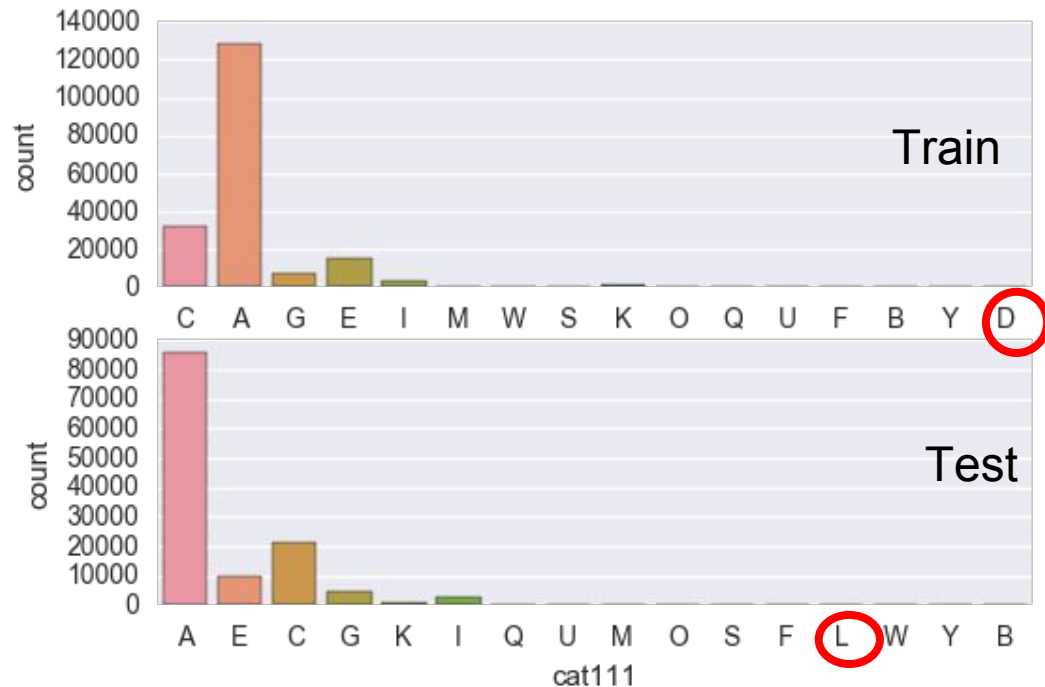
Gray scale image visualising the dataset with features ordered by the log loss. You can see patterns in which the values of particular features change as loss increases.

CAT57 versus Log Loss



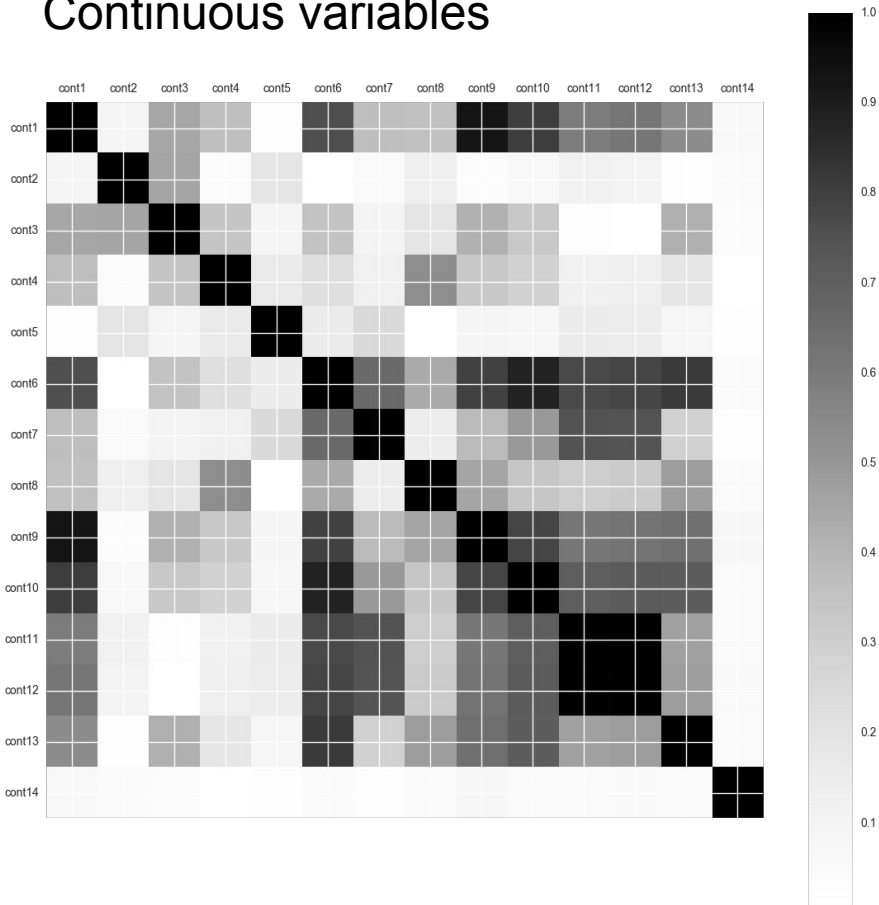
# EDA - Comparison between test and train datasets

Some categorical variables are not present in the test set in total 45 variables



# EDA - Correlation between continuous features

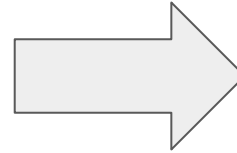
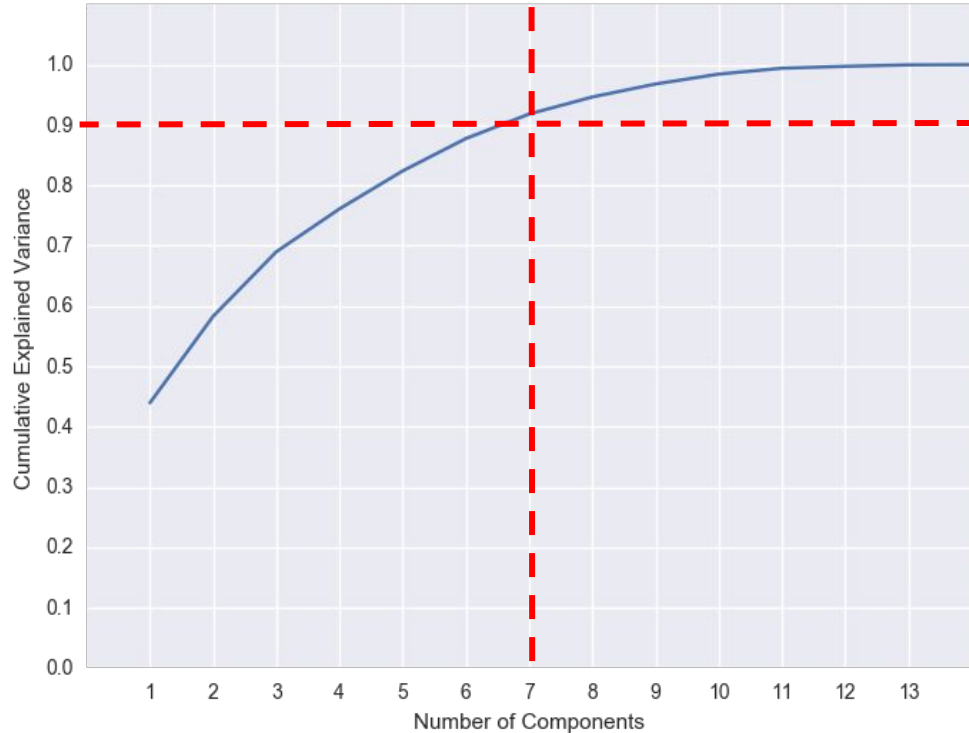
## Continuous variables



Variables	Correlation
Cont 11 & Cont 12	0.994384
Cont 1 & Cont 9	0.929912
Cont 6 & Cont 10	0.883351
Cont 6 & Cont 13	0.815091
Cont 1 & Cont10	0.808551
Cont 9 & Cont 6	0.797544
Cont 9 & Cont 10	0.785697
Cont 6 & Cont12	0.785144

# PCA- Dimensionality reduction

Dimensionality reduction for continuous variables using PCA : We have 14 continuous features



We can reduce the number of continuous data to the half

# EDA - Correlation between 2-variables categorical features

72, 2-variables categorical data (A,B)



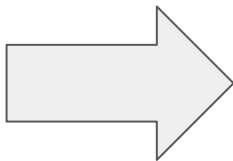
Label encoding



Numerical values

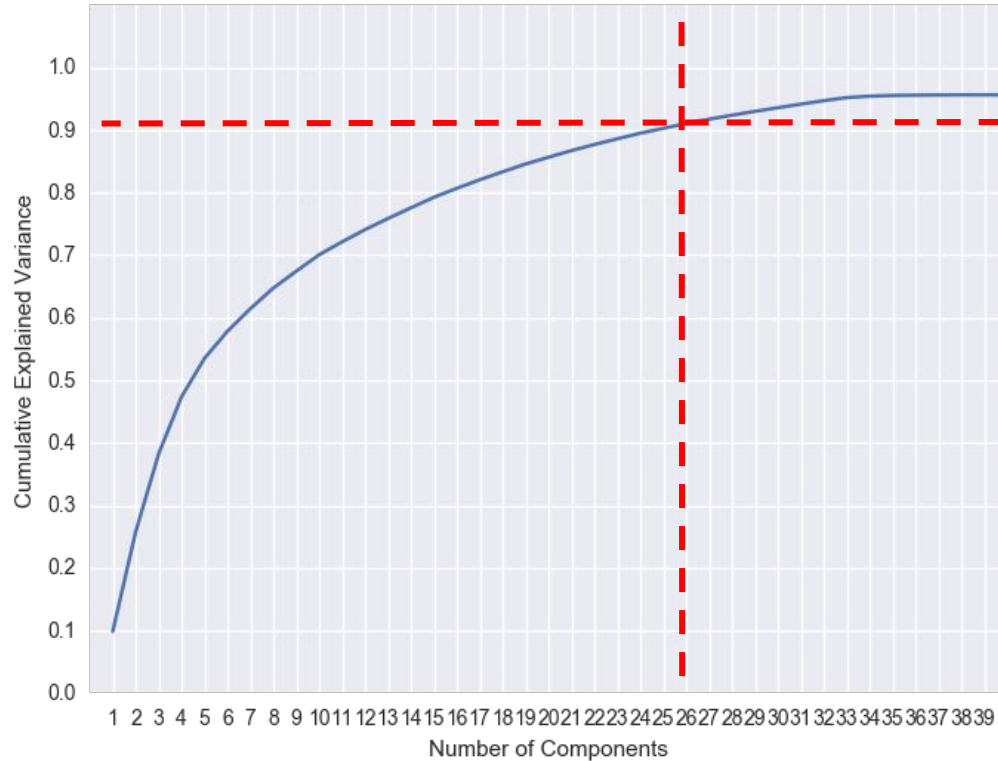


Correlation

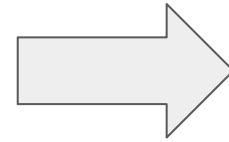


Variables	Correlation
Cat 2 & Cat 9	0.932420
Cat 50 & Cat 6	0.925731
Cat 8 & Cat 66	0.862231
Cont 57 & Cont 7	0.809418
Cont 3 & Cont 16	0.783480

# SVD - Correlation between 2-variables categorical features



SVD: Singular Value Decomposition

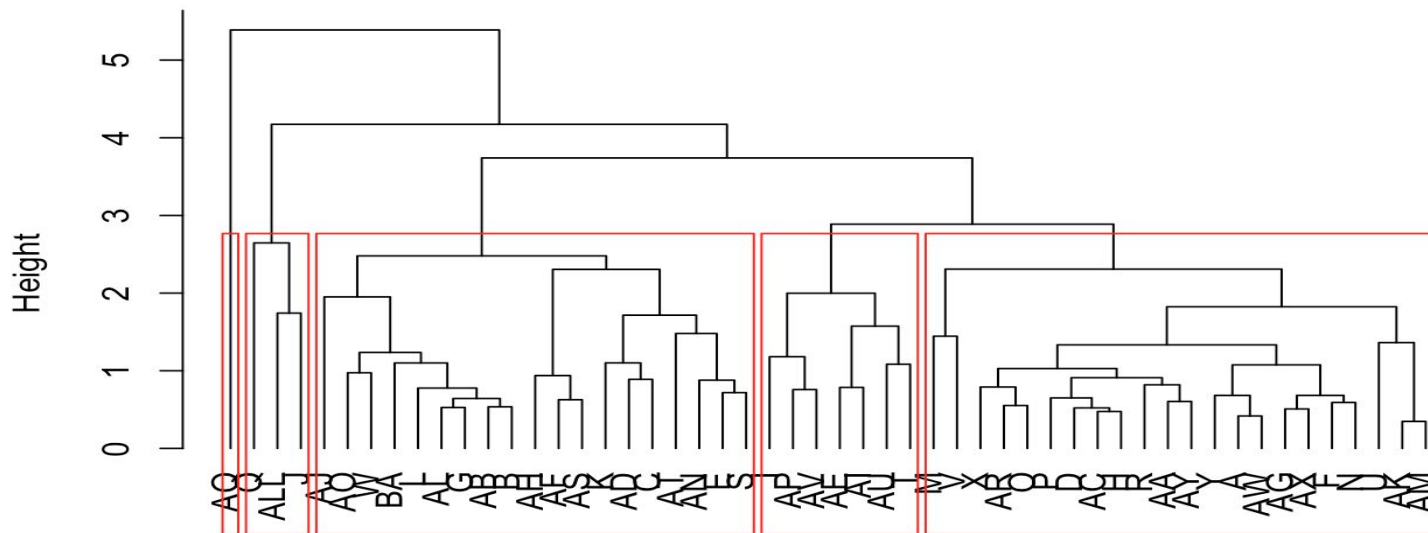


We can reduce the number of 2-variable categorical data to 26 variables



# Hierarchical Clustering into Five Groups

### Dendrogram of Average Linkage 5 Clusters



```
hclust (*, "average")
```

# Hierarchical Clustering Cont...

```
## clusters.average
```

```
## 1 2 3 4 5
```

```
## 3 22 19 7 1
```

```
## cluster Min Q1 Median Mean Q3 Max
## 1 1 23.69 1200 1981.0 2844 3523 121000
## 2 2 123.05 1183 2025.5 2827 3537 24450
## 3 3 189.20 1376 2339.0 3216 4178 30440
## 4 4 201.60 998 1671.0 2428 3015 19620
## 5 5 786.30 1327 2187.0 2735 3123 7998
```

- We can consider performing feature engineering
- We can consider removing AQ - only has 30 observations

# Machine Learning for Prediction

## Tactics to Reduce Iteration Time:

### Regularization

- Near-zero variance function
- Use p values from regression
- Reduced # levels (e.g., cat116)

### Sampling

- Random sampling
- Sampling cat80D versus B

### Other

- Used AWS, but parallel processing not always a turn-key solution
- Reduce # folds in validation

## Models Examined:

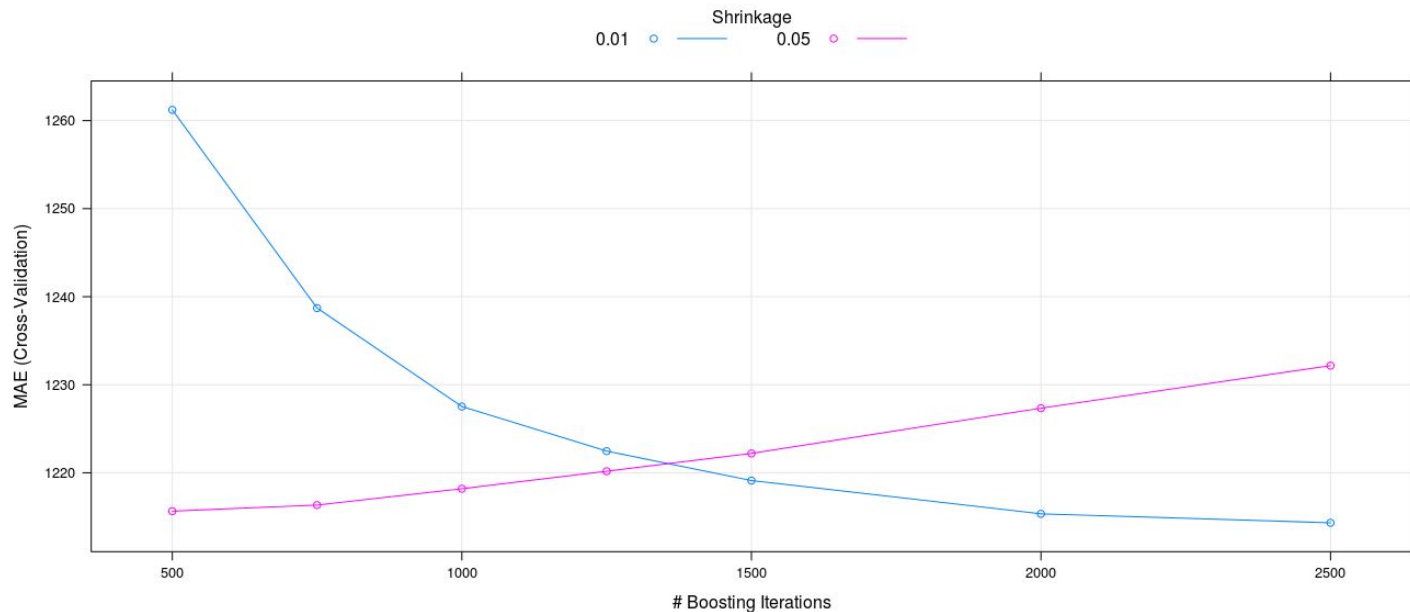
### Regression

- Linear regression --  $R^2$  of 50%. Good for initial analysis
- Boosted trees -- XGBoost had best performance
- Neural network -- close second to XGBoost, but performance lacking on Kaggle test data

### Classification

- Logistic regression
- SVM

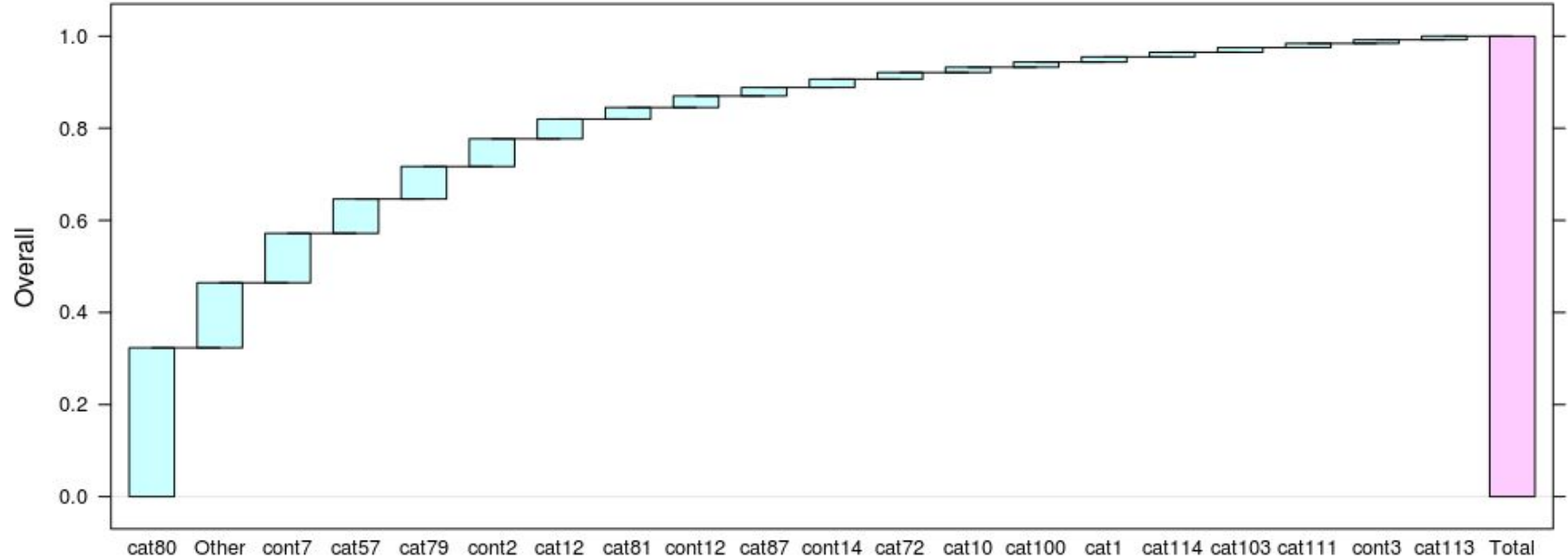
# Machine Learning for Prediction -- XGB Model Tuning



- Final result: 850 iterations, learning rate 0.05, 5 trees
- Prediction unstable with reduced cross-validation folds
- Regularization penalized MAE \$300

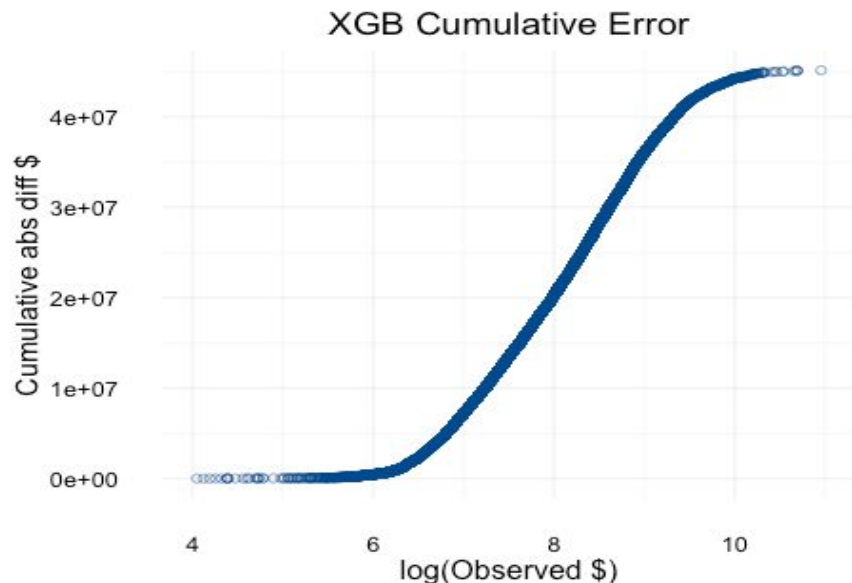
# Machine Learning for Prediction -- Model Assessment

## Variable Importance

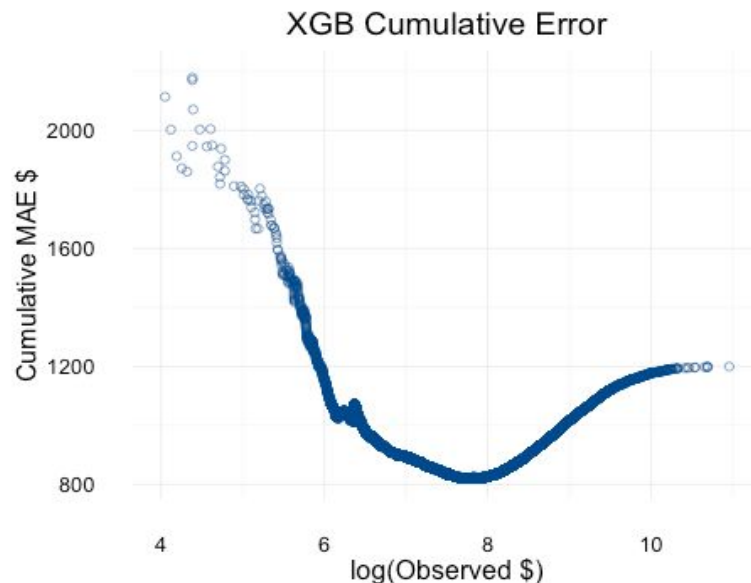


Cat80 largest single predictor

# Machine Learning for Prediction -- XGB Model Tuning

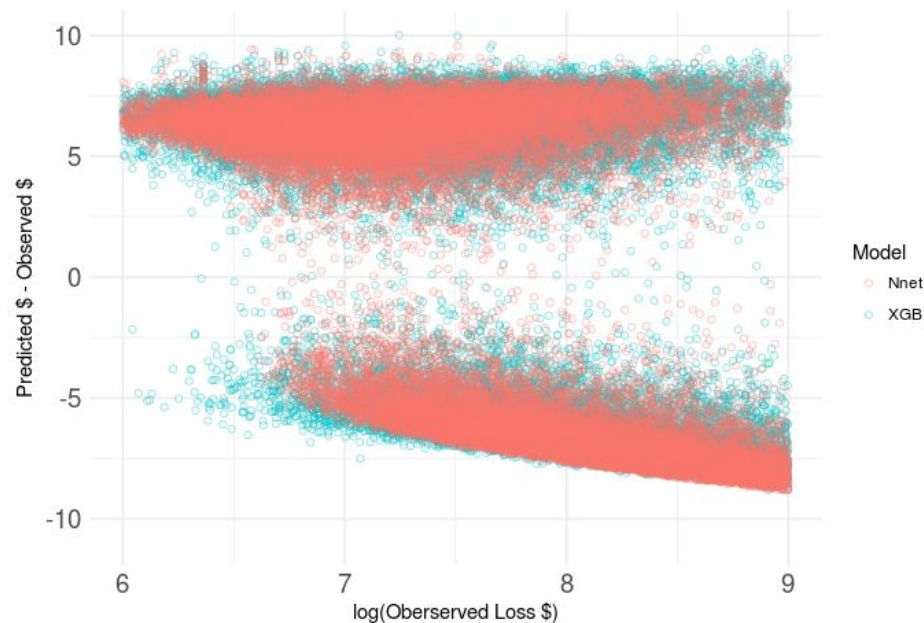
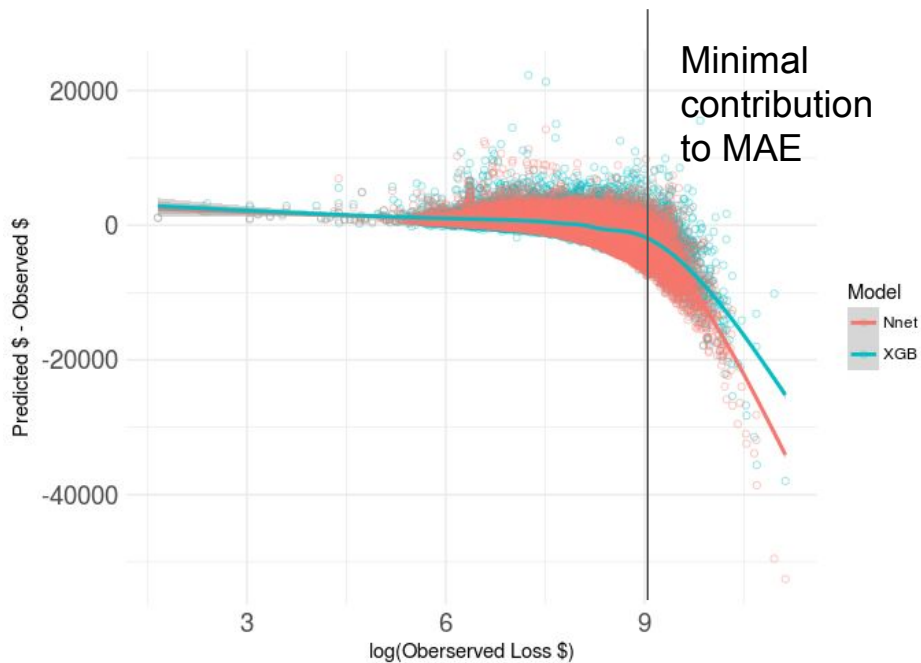


Most of error for claims between  $\exp(\$6)$  and  $\exp(\$9)$  ~(\$400-\$8000), therefore no need to get distracted by tails



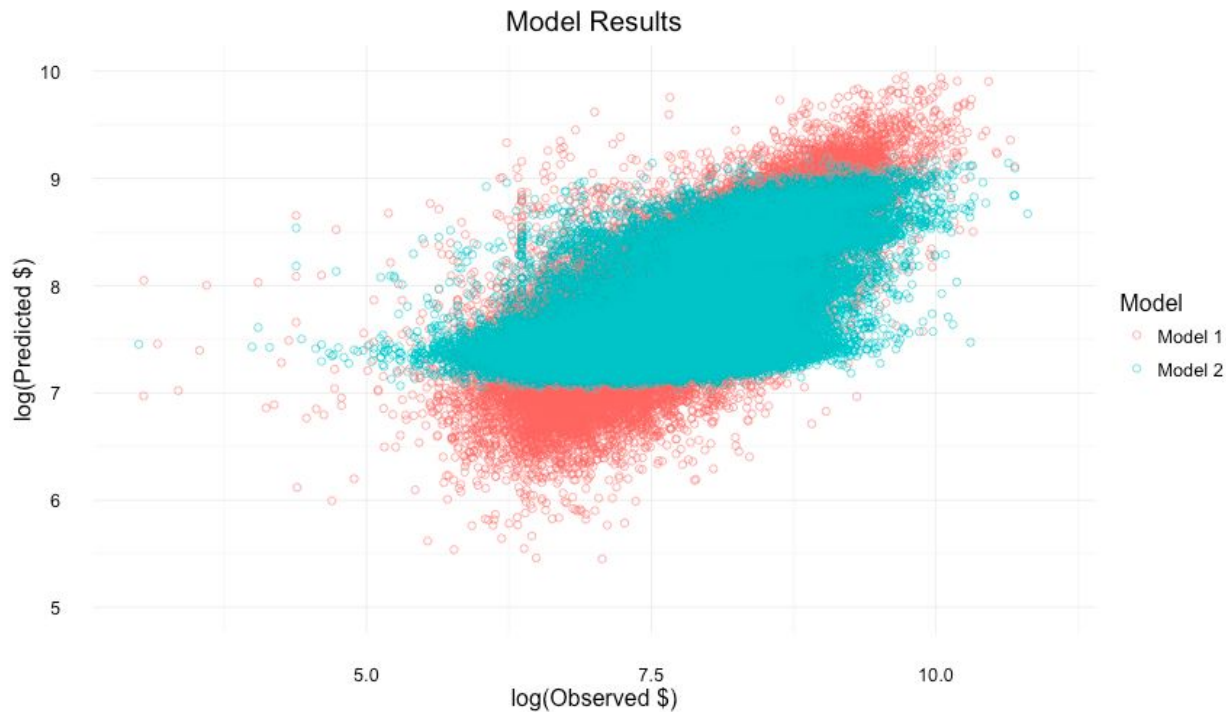
Model gets more accurate until  $\exp(\$8)$  ~\$3000, then performance degrades

# Machine Learning for Prediction -- Model Assessment



Underestimates increase with loss

# Linear Regression w/ 6 Features vs XGBoost with all Features





# What's Salvageable?

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.**

Conclusion: claim size **can not be accurately predicted** based on provided features

## Root Problem

- Doing paperwork for claim **protects insurer** against fraud
- May be able to **reduce paperwork** burden for claims if they don't look "fishy"
- Can features support a classification question?

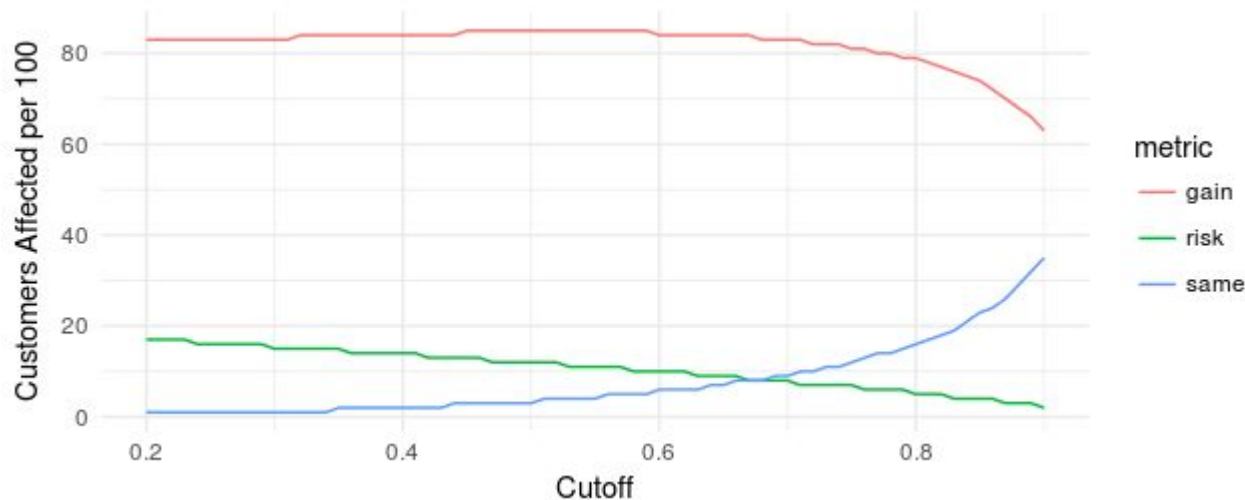
## New classifier: "smallClaim"

- 80% of customers account for 50% of claims by value -- all below \$4500
- "Fishy" claims: feature set associated with small claim value, but requested value is large (the opposite is not important for protection against fraud)

# Fishing for Fishy Claims

Models used: GB, SVM, logistic regression. All had high accuracy but ~50% sensitivity.

We needed something that can be optimized to minimize false positives (logistic)



Next Steps: quantify dollar risk of misclassification and dollar benefit to customer of reduced paperwork

# Classification Model: Low Profile with High Valued Accounts

Confusion Matrix and Statistics		
	High	Low
High	1684	537
Low	1641	13097
<b>Accuracy</b>	0.8716	
<b>95% CI</b>	(0.8664, 0.8766)	
<b>No Information Rate</b>	0.8039	
<b>Sensitivity</b>	0.5065	
<b>Specificity</b>	0.9606	

Training a classification model (using GBM) with a profile having a loss value  $\leq \$4500$  provides 87% accuracy.

Sensitivity (false positive) rate of 50% results in half the low profile accounts being falsely identified as high profile, hence increasing the paperwork process.

Specificity (false negatives) rate of 96% indicates a low error rate of 4%; thus a low probability of missing a low profile account with a high value claim request.

# Performance Metrics

State	cat112 == E	cat112 == E, AH	cat112 == E, AH, AS
nrow	25148	43787	61456
%	13.35%	23.25%	32.63%
<b>LM - Linear Models</b>			
mean	0.5972342	0.5838424	0.5812039
RMSE	0.586739	0.5869689	0.5799521
<b>GBM - Generalized Boosted Regression Models</b> <i>An implementation of extensions to Freund and Schapire's AdaBoost algorithm and Friedman's gradient boosting machine.</i>			
mean	0.5794056	0.5588792	0.5567978
RMSE	0.5693272	0.5648724	0.5527663
Score	X	X - 154.32	X - 42.75

# Conclusion

## Kaggle Competition

- More data doesn't improve predictive power
- Continue to fine tune models to improve scoring (mean absolute estimate) in Kaggle

## Business Insight

- Tune classifier model by incorporating missing categorical values into the training / test set
- Reducing CV folds is a mixed blessing: useful initially, but becomes easy to overfit