Predicting Insurance Claims in Brazil

Dixee Kimball (dkimball@stanford.edu), Laura Zhang (lzhang96@stanford.edu), Matthew Millican (millimat@stanford.edu)
CS 229, Autumn 2017

Introduction

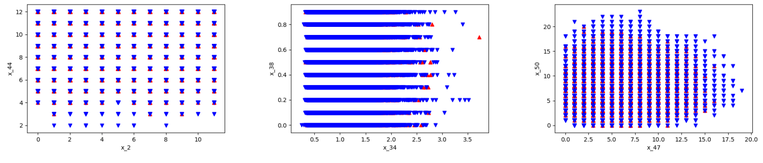
Accurately predicting the likelihood of a customer filing an insurance claim increases car-ownership accessibility for better drivers and allows car insurance companies to both charge fair prices to all customers and increase profits. Given **unlabeled features about a customer,** can we **predict whether the customer will file an insurance claim** during a period of interest?

Data

We analyzed a dataset of 595,213 customers of Porto Seguro, one of Brazil's largest auto and homeowner insurance companies. Each record $x^{(i)}$ consists of $n = 57$ unlabeled but distinct features, and a label $y^{(i)} \in \{0, 1\}$ indicating whether customer i filed a claim.

Notable characteristics:

- 21,694 (3.6%) examples have label 1, and the remaining 573,518 (96.4%) have label 0
- All features are unlabeled to protect the identity of Porto Seguro's insurance customers
- Data does not display obvious linear dependencies; see figures below



Baseline

Our baseline model **generates labels from a Bernoulli random process** with the probability of filing a claim calculated as the **mean percentage of customers with a claim:**

$$p = \frac{1}{m} \sum_{i=1}^{m} y^{(i)}$$

This model achieves a *normalized Gini score* of $4.14 \cdot 10^{-4}$.

Models

We utilized the following models to predict probabilities that customers would submit claims:

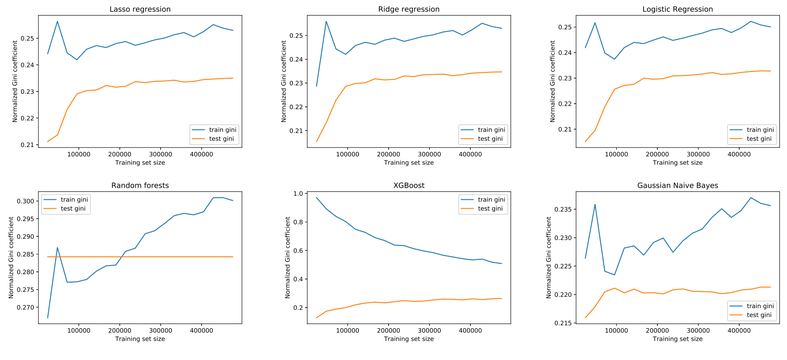
- **Lasso regression:** Least-squares with L^1 regularization $\|\theta\|_1$
- **Ridge regression:** Least-squares with L^2 regularization $\|\theta\|_2^2$
- **Logistic regression:** Output claim probability $\sigma(\theta^T x)$
- **Random forests:** Combine results from multiple decision trees
- **Gradient boosting:** Decision trees fitted to residual errors of previous trees
- **Gaussian Naive Bayes:** Assume features distributed normally given class labels

Models were evaluated using a *normalized Gini coefficient:*

1. Use predicted labels as keys to sort ground-truth labels
2. Compute Lorenz curve over sorted ground-truth labels
3. Compute Gini score for Lorenz curve
4. Divide by Gini score of a perfect classifier

Training

The below learning curves for each model demonstrate model performance as a function of training set size (linearly spaced from 0 to 476,169 examples). The test set size is kept constant (119,043 examples).



Evaluation

Kaggle hosts an additional dataset for model evaluation (892,816 examples) Submitting our predictions yielded the following normalized Gini scores:

Model	Train Score	Test Score	Kaggle Submission Test Score
Lasso	0.253	0.235	0.251
Ridge	0.253	0.235	0.251
Logistic Regression	0.250	0.233	0.249
Random Forests	0.300	0.284	0.245
XGBoost	0.508	0.263	0.276
Naive Bayes	0.236	0.221	0.235

Discussion

- **Most models generalize** from training set to test set
- **Linear models performed better than expected,** especially given that linear relationships in the data are not immediately obvious
- **Random Forests did not perform as well as expected,** which may mean ensemble methods are not as promising
- XGBoost **overfit the training set** yet **performed the best on the Kaggle test set**

Future Work

- Neural networks
- Continue hyperparameter search for XGBoost
- Ensemble existing learning methods
- Impute missing feature values with sample means