Regressing Loan Spread for Properties in the New York Metropolitan Area

Tyler Casey
tyler.casey09@gmail.com

Abstract:

In this paper, I describe a method for estimating the spread of a loan given common quantities filed with the loan itself. Estimating the risk and reward of a loan has been of particular importance since the economic crisis of 2008, as a more accurate estimation and validation scheme on loan risk could have helped prevent and mitigate the widespread damage of the mortgage and derivatives crisis. The dataset used in this analysis was a small group of recent (2013) loans originating in the New York Metropolitan Area. The small sample of loans available, mixed with a large feature space and multiple loan types within the dataset itself, made creating a robust model primarily a task of avoiding high-variance. Overall, I achieved a substantial improvement over an average baseline estimate, but acknowledge room for changes in the model and recognize some practical limitations with the analysis.

Introduction:

The financial crisis of 2008 has been largely attributed to a dramatic rise in defaulting home loans, and a collapse in credit derivatives created from bundles of bad loans [1]. Pricing loans to effectively cover their true risk of default is axiom in the financial world. The volume and value of both loans and their derivative assets is a strong impetus for creating a programmatic approach to both pricing and validation. The complexity in the current task is broken into three phases: data preparation, feature enhancement and selection, and model generalization. All coding was done in Python, utilizing the machine learning package Scikit-Learn [2].

Data Preparation:

Individual loan data was procured from Trepp.com, a mortgage securities data reseller, by the project Sponsor. For the analysis the dataset was confined to recent loans on properties in the New York Metropolitan Area. Additionally, loans were filtered so that each was guaranteed to be a single-loan property, benchmarked by US Treasury notes, and have a LTV (Loan to Value) measure. For the analysis, all non-numeric columns in the data were removed. In future analysis it could prove useful to encode these text fields as

features, but given the already large number of variables per sample and the possible misinterpretation of the text data itself, this was left out. Samples were then binned on property type (i.e. Multifamily, Hotel, Office, etc.), and all columns were L2 normalized (L1 and unnormalized data was also tried, but were found to be in the Best Fit models detriment). Properties were binned because loan providers consider certain property types more risky than others.

The dependant variable for the analysis was the average of a loan's low and high spread. This number represents the premium the bank stands to make over the current lending rate from the Federal Reserve, and is the measure of risk for a loan.

Histogram Binning

At the suggestion of the data provider, a histogram binning approach was done to create a piecewise linear model for capturing different phases within important features, namely the Loan to Value (LTV) metric. Instead of doing this manually, I implemented a tunable algorithm for identifying regions of significance by convolving the cumulative distribution of a variable with a 2nd-derivative gaussian. The result being a smoothed version of the second derivative of the CDF. A heuristic binning strategy was applied on top of this within each property type. Examples of dynamic binning on the LTV metric for two different property types below in figure 1.

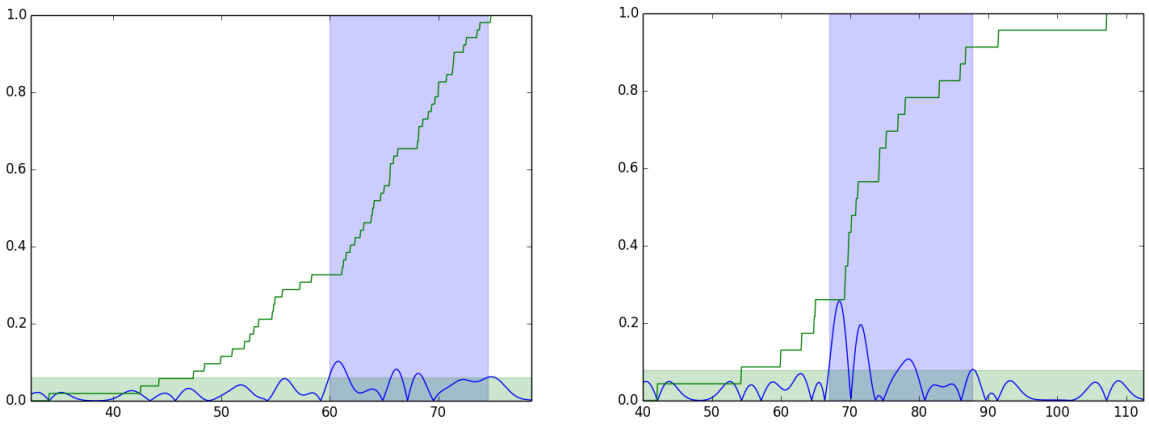

Fig. 1: The thin blue line is the smoothed 2nd-derivative of the data, the green bar represents a tunable threshold for making a ranging number of bins. The green line is the CDF of the LTV for the property type in question.

Census Data Extraction:

An effort to provide further context for the loans beyond the Trepp data was done by accessing census information on demographics for each property in the data set. This was done via the publically available Cdyne website [3]. Rate limiting on the free information required building a worker cluster to facilitate the full request in reasonable time.

Worker Cluster:

In order to extract the Census data, and later to do a large scale hyperparameter search on regression algorithms, I procured a distributed cluster of 10-50 linux workers on cloud services platform Heroku, along with a small instance running the Redis database. Using the task distribution package Celery, tasks were dispatched to workers and then processed in batch after completion.

After adding census data and binarizing important variables, the final data set is comprised of 145 loans, with a maximum of ~80 features (mostly binary variables), depending on the settings of the histogram binarizer.

Feature Selection and Model Selection:

Given the number of samples compared to features, narrowing the feature set to reduce the risk of high variance models was a priority. Initial tests with various linear models displayed high variance on hold out cross validation sets. Ensemble regression algorithms address this problem by combining multiple candidate models to form a more generalized complete model. The regressors ultimately up for fine tuning were Random Forest Regressor, and Extra Trees Regressor [4, 5] . These tree based ensembles are useful for both regression and for feature selection, as the algorithms must score many sub-regressors on limited feature sets during the fitting process, creating a built in measure of feature importance, figure 2 below shows the top 10 features for the project's best fit.

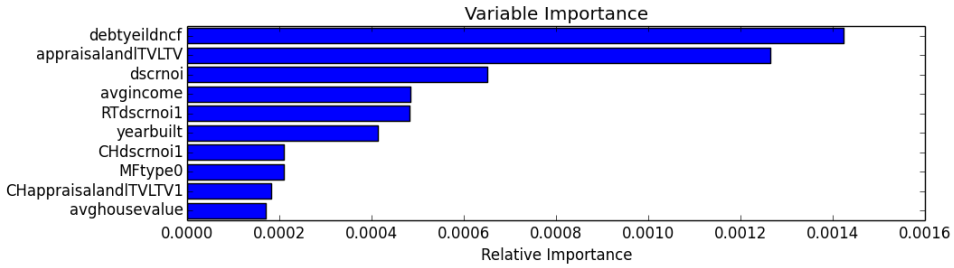

fig 2.

VC Dimension and Performance Measurement:

A literature search into the VC dimension of ensemble regressors was inconclusive as to how the properties of a data set effect error estimates. Intuitively, the number of binary dimensions provided by the binning functions (~30 per variable binned) translates into a large generalization error for less complex algorithms given the size of the dataset. Figure 3. below illustrates the substantial test error in practice on a K-Fold=20 Random Forest analysis with increasing number of Trees in the regressor.

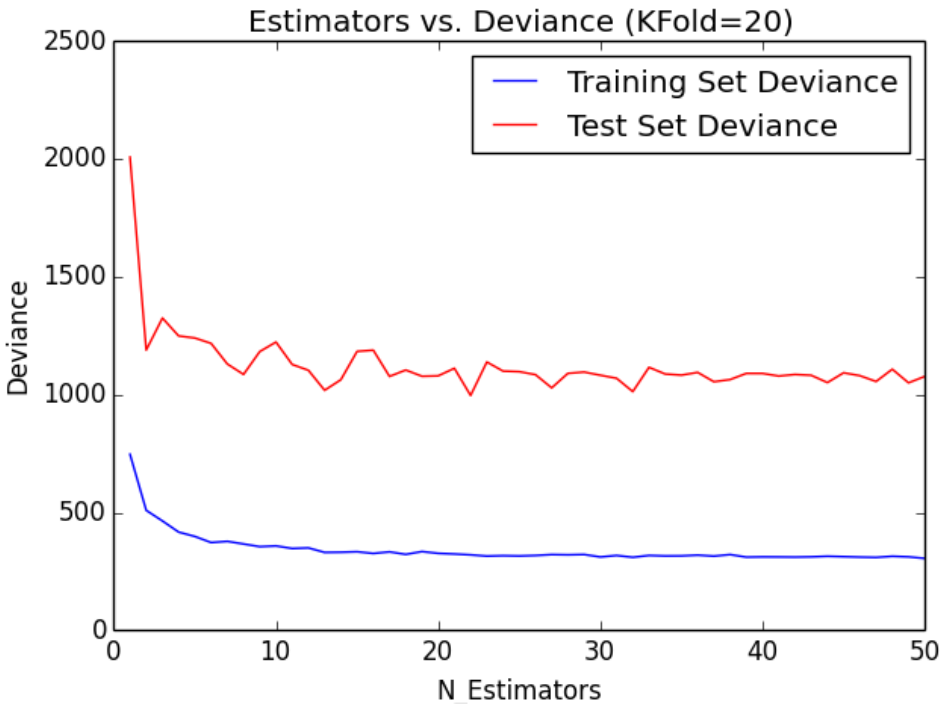

fig. 3

With this in mind, In order to maximize the training set size, model performance was gauged on N Leave One Out validation steps, with an output metric of Mean Squared Error (MSE).

Hyperparameter Search:

In order to fine tune the ensemble regressors using Leave One Validation, which is costly in practice, a hyperparameter search was done on the valid parameters of the algorithms. A powerset of viable ensemble parameters and feature sets was dispatched to the worker cluster. Model fits were tested and MSE measures were stored in Redis. Approximately 35000 parameter sets were tried in a 12 hour period.

Best Fit Results:

Data Parameters:	***Regressor Paramaters:***
Bins: ['appraisalandlTVLTV', 'dscrnoi'] Features: ['married', 'avgincome', 'avghousevalue', 'yearbuilt', 'debtyeildncf', 'appraisalandlTVLTV', 'adjaveragespread', 'dscrnoi'] Columns L2 Normalized: True	Algorithm: RandomTreesRegressor N_estimators: 100 Min_samples_leaf: 2 Min_samples_split: 3 Max_features: 6

Best Fit Regressor MSE: 800.4
Data Average Dummy Regressor MSE: 1620

Discussion and Conclusion:

Although many things were tried in this project to reduce the MSE of the test sets, frustratingly it seemed as though the features of the data did not have much informative capacity given the number of samples. This shelved some of the later analysis planned for this project, and is a tentative critique of the information associated with loans, i.e. it is seemingly of little value in inferring a loan's risk. I was able to cut the baseline MSE by a factor of two, which is a substantial yet somewhat lackluster improvement. Nonetheless, the error difference between a dummy regressor and the best fit model corresponds to approximately 10 spread points. Considering the value of the financial assets in question, a better regression by 10 spread points could translate to significant efficiencies at large scale.

Citations:

[1] "Financial Crisis of 2007–08." *Wikipedia*. Wikimedia Foundation, 12 Sept. 2013. Web. 12 Dec. 2013. http://en.wikipedia.org/wiki/Financial_crisis_of_2007

[2] "Scikit-learn." *: Machine Learning in Python — 0.14 Documentation*. N.p., n.d. Web. 14 Dec. 2013. http://www.scikit-learn.org

[3] "Demographic Data." *Demographic Data*. N.p., n.d. Web. 14 Dec. 2013. http://www.cdyne.com/free/demographic-data.aspx

[4] L. Breiman, "Random Forests", Machine Learning, 45(1), 5-32, 2001.

[5] P. Geurts, D. Ernst., and L. Wehenkel, "Extremely randomized trees", Machine Learning, 63(1), 3-42, 2006.