CSE353 Final Repot

Team Value Assess Discrepancy Regression (VADR)

Anthony Musco, Matthew DelSignore, Gustavo Poscidonio, Fumi Honda

Abstract

The goal of this project is to produce a classification algorithm capable of determining whether or not a particular property is over-assessed or under-assessed. This type of classifier would be very useful for professionals in the real estate industry in helping to determine if the current level of taxation for a particular property is appropriate.

Introduction

The taxation of commercial real estate is an immensely important process that impacts property owners and municipalities alike. In New York City, nearly 40\% of the City's total budget is generated through property taxes, producing more than $18 billion in revenue annually. Despite the huge numbers involved, errors in taxation exist throughout the tax base, and property owners regularly pay too much or too little in taxes every year. This situation occurs when the true market value of a property appreciates or depreciates over time without a proportional adjustment in the assessment. These discrepancies accumulate and are typically only corrected after municipality-wide 'revaluations', or 'revals', or if the property owner successfully completes an appeal to have his/her assessment reduced.

Definitions and Terms

Property/Tax Lot: Well-defined area of land within some municipality (City of New York), usually indicated by some form of identification number (Borough, Block , Lot). Tax lots are the fundamental constituents of real estate transactions, and typically have records to describe various features of the property which sits on top of it (building area, building age, date of last sale, etc.)

Market Value: The amount of currency an unencumbered buyer would be willing to pay an unencumbered seller in exchange for the property in question, assuming an arm's-length, open-market transaction. It is assumed that neither the buyer nor the seller has any strong reason to buy or sell respectively, and that no other conditions affect the terms of the exchange.

Assessment: The value assigned to a tax lot by a governing body (City of New York) to used to calculate the annual property tax. Note that although the assessment is intended to have a direct relationship with market value, the two are not the same and are often wildly different.

Property Tax: Annual tax on a property levied by the governing body. Property taxes are directly related to the assessment, and are typically given by the equation *tax = assessment * rate*. Therefore, a high assessment indicates a high property tax, and a low assessment indicates a low property tax.

Data Source

There are approximately 1.1 million properties located within New York City. On average, just over 40,000 of these properties are exchanged every year, with a majority of them being exchanged at or around market value. In order to regularize the sample set, we eliminated all properties except for those in **Tax Class 2, Building Class C1**. The properties that remained are multifamily residential (apartment) buildings, which have historically been stable in market value due to their non-volatile income streams, tax benefits, and relative ease of financing. From this pool we removed any extreme outliers from this pool (i.e. price/unit, price/sqft > mean ± 3 * st.dv.), and collected all properties which have sold within the past 4 years (Jan-2011 - May-2015), leaving us with exactly 1,109 data records.

Classification Scheme

1. Over-Assessed 1
2. Under-Assessed -1
3. Appropriately Assessed 0

The first challenge is defining what is actually meant by 'over-assessed' and 'under-assessed'. In most municipalities, assessments of properties are designed to be significantly below actual market value, for statistical and legal reasons. In New York, however, we are given a much better representation of the assessment, which is the **Indicated Market Value.** This is the value which the City of New York uses to derive the assessment, and is designed to be equal to exactly 45% of the actual market value. Therefore the feature used to classify our training set is the *ratio* of the indicated market value and the actual market value. The following classification function is used to classify the data, where *t* represents our threshold value, which is the number of standard deviations away from the mean after which a property can be considered over-assessed or under-assessed:

$$\text{Class}(\mathbf{x}^{(I)}) = \begin{cases} 1 & \frac{\mathbf{x}^{(I)}_{ind}}{\mathbf{x}^{(I)}_{amue}} \geq \mu_r + \tau \cdot \sigma_r \\ -1 & \frac{\mathbf{x}^{(I)}_{ind}}{\mathbf{x}^{(I)}_{amue}} \leq \mu_r - \tau \cdot \sigma_r \\ 0 & \text{otherwise} \end{cases}$$

Methodology

Raw data was acquired from New York City Department of Finance's web page. Although most of the data could be downloaded as Excel spreadsheets, a few of the features for each record needed to be scraped from the web, which was accomplished via a Python web-scraper. The filtering and classification described above was done in Excel, which was exported to a comma-separated value format. In addition to the standard output file, a normalized file was also produced, which included normalized features (each feature subtracted from its mean and divided by its standard deviation), which is much easier to learn on. The data was then separated into two sets - training data (900 records) and test data (209 records).

After collecting and cleaning the data, several different classifiers were tested to attempt to generalize the relationship between the features and its relative assessment. Each classifier was cross-validated with varying degrees of transformed features to see which degree produced the best results. After each degree was selected, parameters were cross-validated against a fixed degree to see which model parameters generated the best results. The best degree-parameter pair was selected for each model, and the best model overall was selected as the classifier.

Feature Space

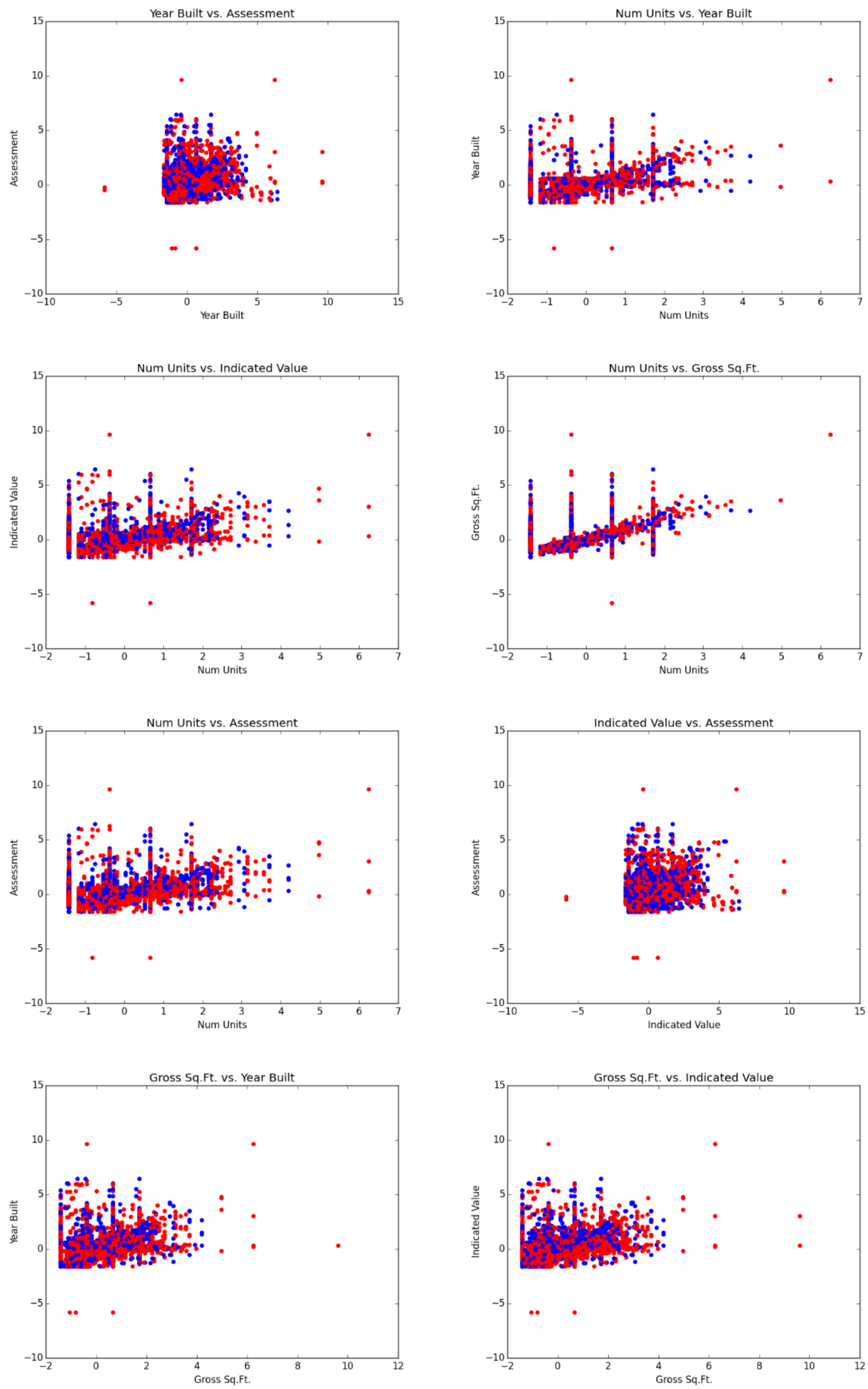
The following features were used for learning, which had the statistical distribution indicated by the table below.

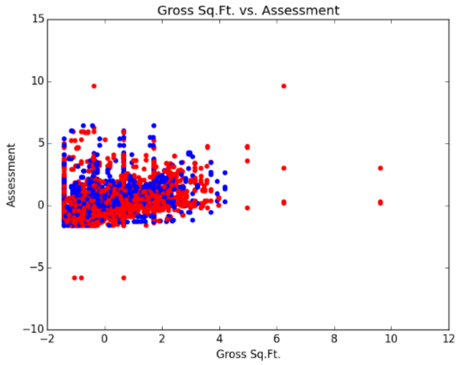
1. Borough - CATEGORICAL { Manhattan-1, Bronx-2, Brooklyn-3, Queens-4 }
2. Units - INTEGRAL { ≥ 5 }
3. SqFt - INTEGRAL { ≥ 0 }
4. Year - INTEGRAL { 1800 - 2015 }
5. Ind. Value - INTEGRAL { $\geq$ 100,000 }
6. Assessment - INTEGRAL { $\geq$ 5,000 }

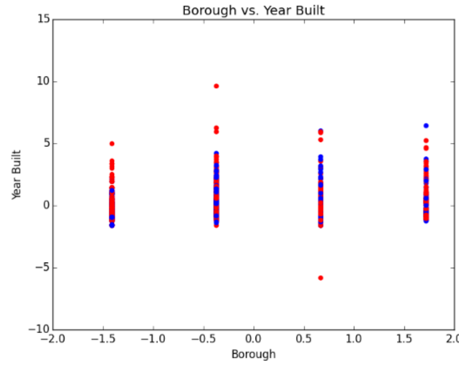
Statistic	Borough	Total Units	Gross Sq.Ft.	Year Built	Ind. Value	Assessment
Minimum	1	11	3,750	1840	178,000	7,200
Maximum	4	116	153,952	2014	7,884,000	3,425,940
Mean	2.35	27.58	22,507.60	1922	1,113,765	402,410
Standard Deviation	0.96	14.22	13,761.29	14	709,303	274,628

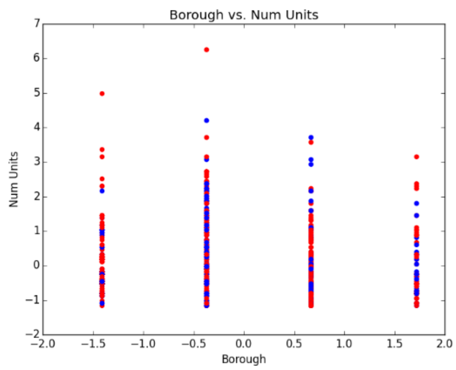
Data Separability

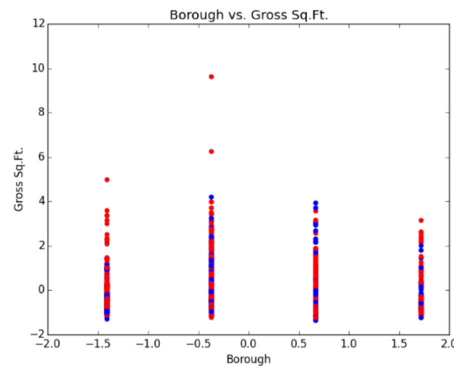
The following set of graphs plot each and every feature against another as well as against the assessed value. We tried to use these graphs to determine if, on some dimension, the data is linearly separable. Through these graphs, we are able to observe that there are some linear trends in the data, however, no two dimensions show the data to be linearly separable.

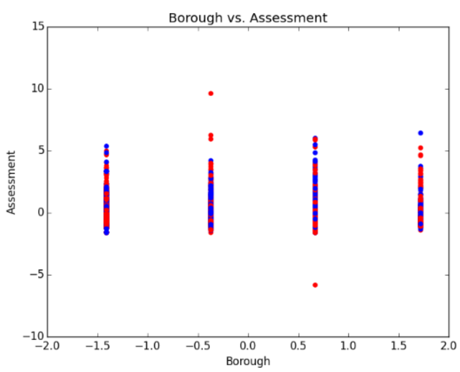

Year Built vs. Assessment
Assessment
Year Built
Num Units vs. Year Built
Year Built
Num Units
Num Units vs. Indicated Value
Indicated Value
Num Units
Num Units vs. Gross Sq.Ft.
Gross Sq.Ft.
Num Units
Num Units vs. Assessment
Assessment
Num Units
Indicated Value vs. Assessment
Assessment
Indicated Value
Gross Sq.Ft. vs. Year Built
Year Built
Gross Sq.Ft.
Gross Sq.Ft. vs. Indicated Value
Indicated Value
Gross Sq.Ft.

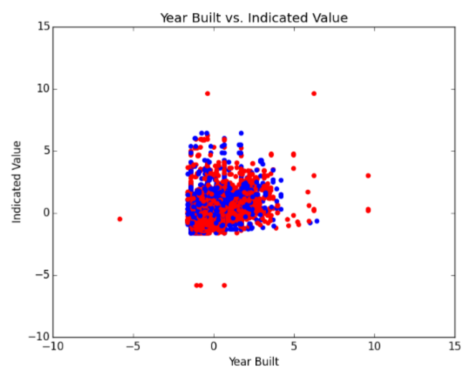

Gross Sq.Ft. vs. Assessment
Assessment
Gross Sq.Ft.


Borough vs. Year Built
Year Built
Borough


Borough vs. Num Units
Num Units
Borough


Borough vs. Gross Sq.Ft.
Gross Sq.Ft.
Borough


Borough vs. Assessment
Assessment
Borough

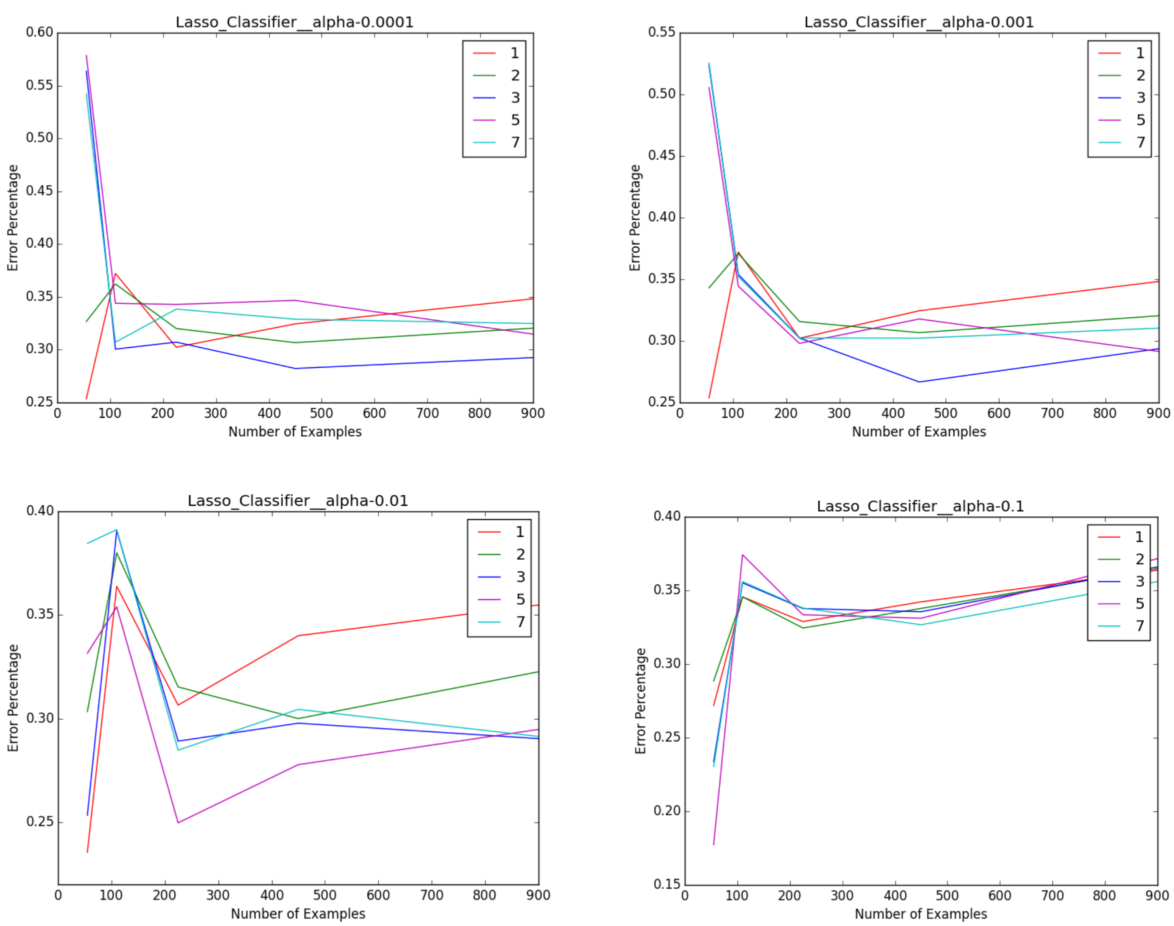

Year Built vs. Indicated Value
Indicated Value
Year Built

Classifiers

LASSO Regression

A LASSO classifier was applied to the data with several fixed alpha values to determine the learning curves for various degrees of transformation for the feature set. The cross-validated learning curves for alpha values {1e-4, 1e-3, 1e-2, and 1e-1} are shown below:

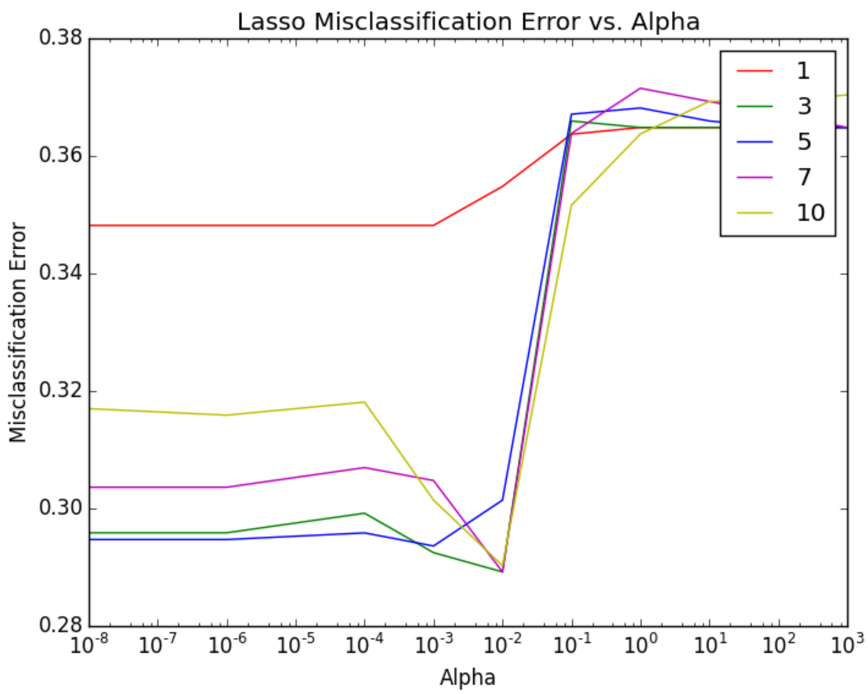
Learning Curves



As can be seen, the most consistent performance was achieved with a transformed feature space of degree 3. It is also easy to note that the optimal performance seems to settle between 30%-35% misclassification rate, regardless of the degree. This might indicate a theoretical limit to performance based on uncertainty and noise in the data.

Parameter Tuning

Generating a LASSO classifier based on a range of alphas from [1e-8, 1e3] produced the following plot:

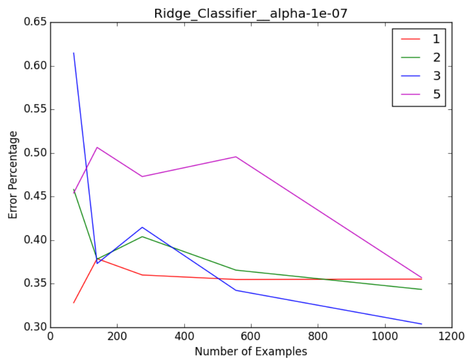


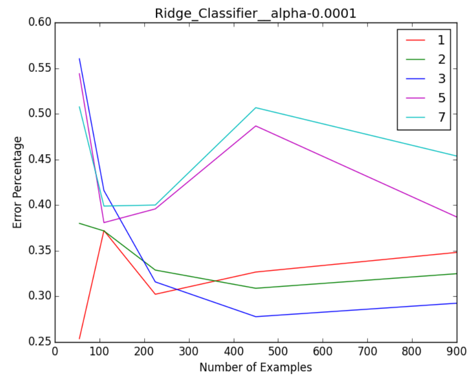
Consistent with the learning curves above, degree 3 performed the best, and generated the best results with an alpha between 1e-1 and 1e-2. The misclassification rate for this classifier was approximately 29%, which seems consistent with other classifiers tested in the project.

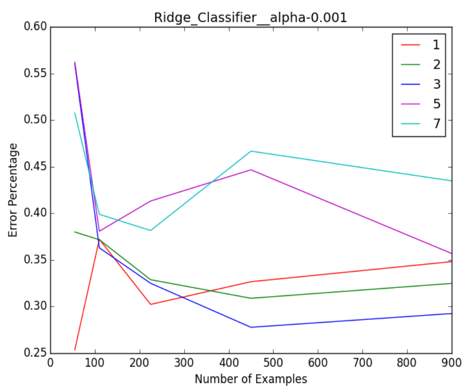
Ridge Regression

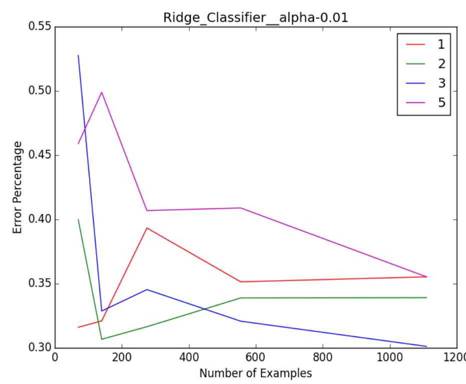
A Ridge classifier was applied to the data with several fixed alpha values to determine the learning curves for various degrees of transformation for the feature set. The cross-validated learning curves for alpha values {1e-7, 1e-4, 1e-3, 1e-2, 1e-1 and 1e0} are shown below:

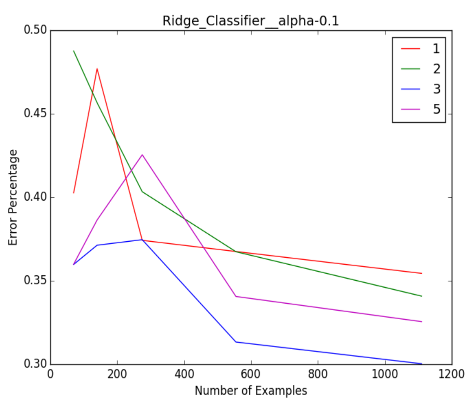
Learning Curves

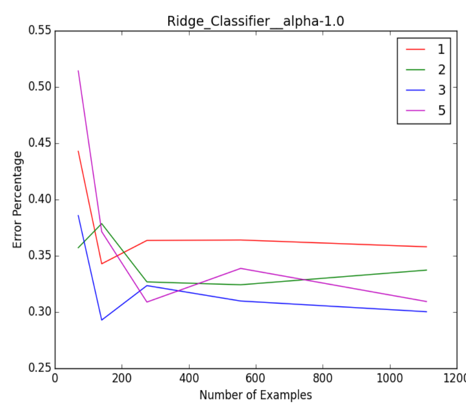








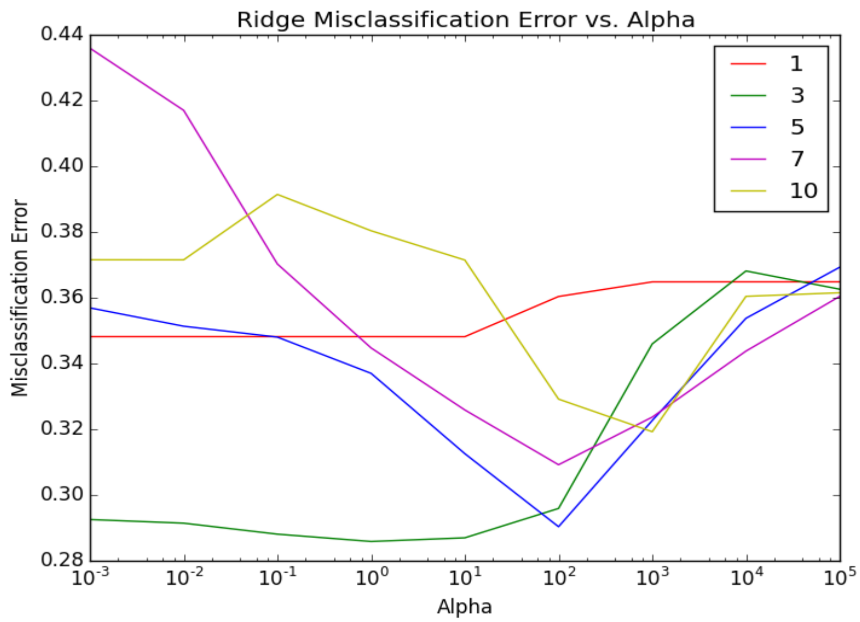




Once again we observe that degree 3 polynomials consistently achieve the smallest misclassification error. We the misclassification rate approaching 25%-30%, performing similarly to LASSO regression, further suggesting the aforementioned theoretical limit to performance.

Parameter Tuning

Generating a Ridge classifier based on a range of alphas from [1e-3, 1e5] produced the following plot:

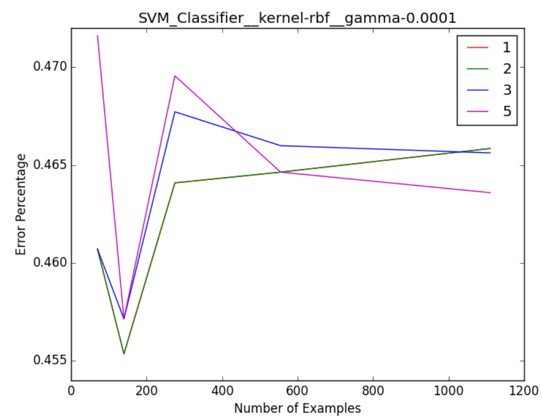


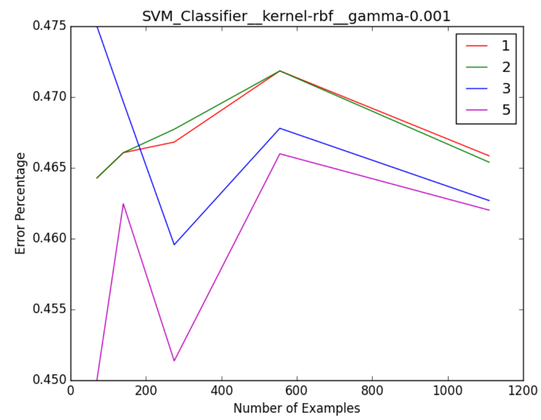
Consistent with the learning curves above, degree 3 polynomials achieved the best performance with the lowest misclassification rates occurring with an alpha of 1e0 at approximately 29%.

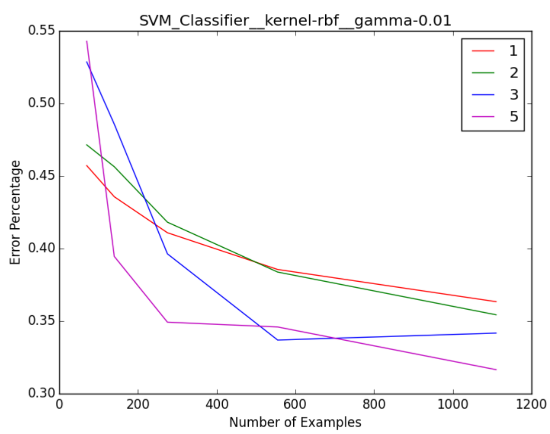
SVM - RBF Kernel

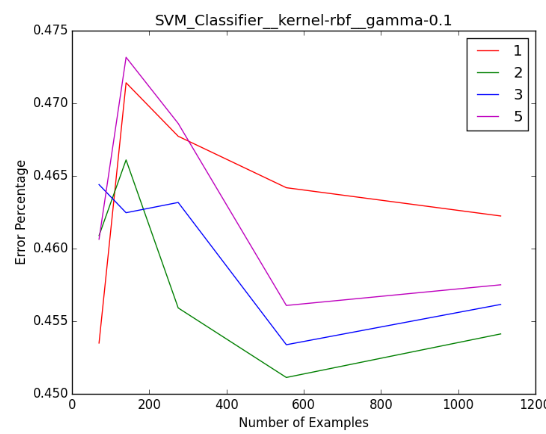
A Support Vector Machine with an RBF Kernel was applied to the data with several fixed gamma values to determine the learning curves for various degrees of transformation for the feature set. The cross-validated learning curves for gamma values {1e-4, 1e-3, 1e-2, 1e-1 and 1e0} are shown below:

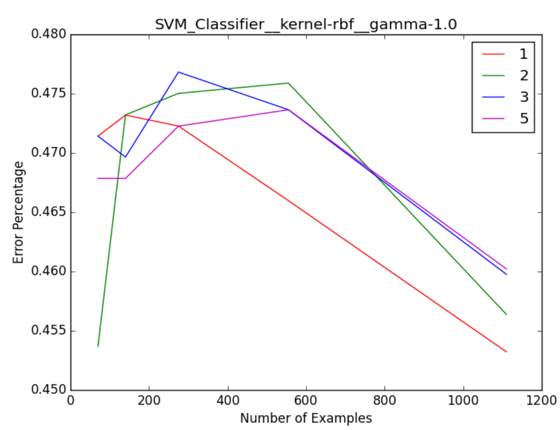
Learning Curves







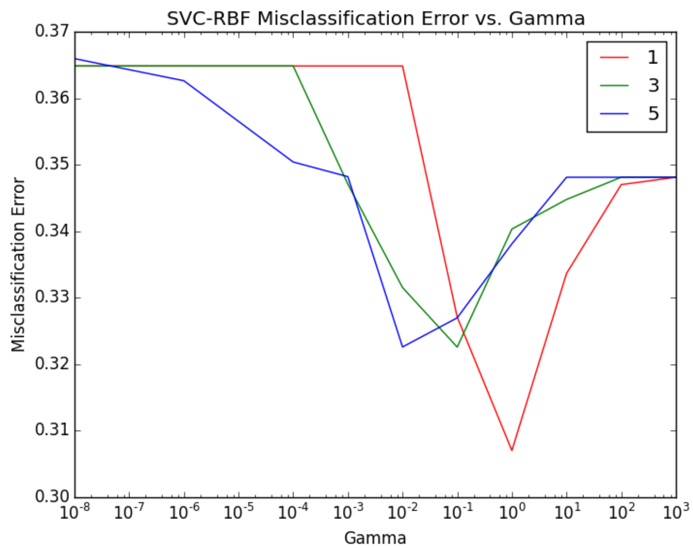




Compared to LASSO and Ridge classifiers, SVM – RBF Kernel performed significantly worse and inconsistently so. Depending on the gamma, polynomials of different degrees performed better so no true conclusion can be drawn as per the best degree of a classifying polynomial. Furthermore, on average, the different polynomials approached a misclassification rate of 45%.

Parameter Tuning

Generating a SVM – RBF Kernel classifier based on a range of gammas from [1e-8, 1e3] produced the following plot:

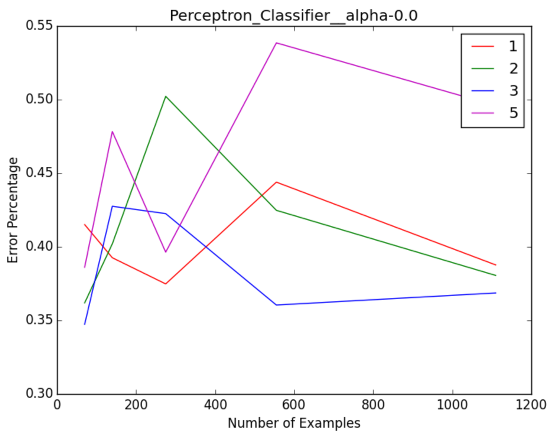


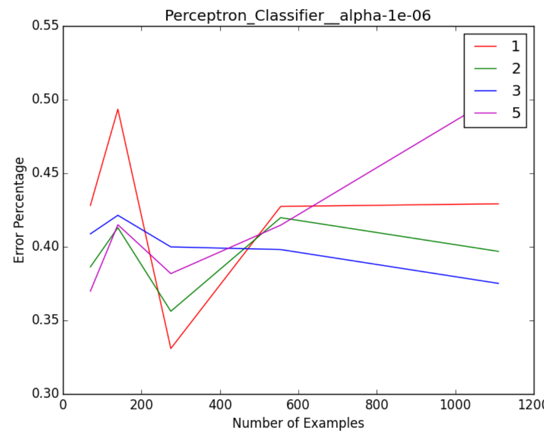
Evidently, an SVM – RBF Kernel produces inconsistent and inferior results when compared to the LASSO and Ridge classifiers. Even with the most promising degree and gamma value, we still can only achieve a 31% misclassification error.

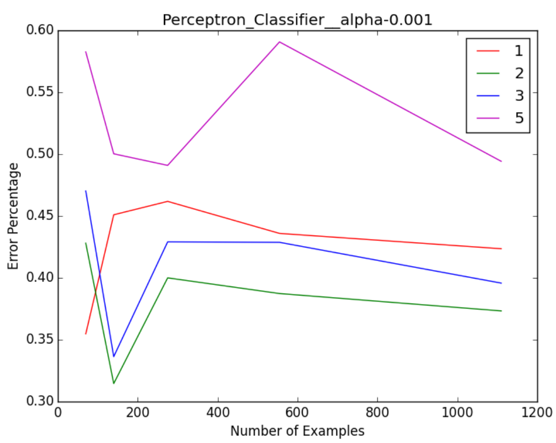
Perceptron

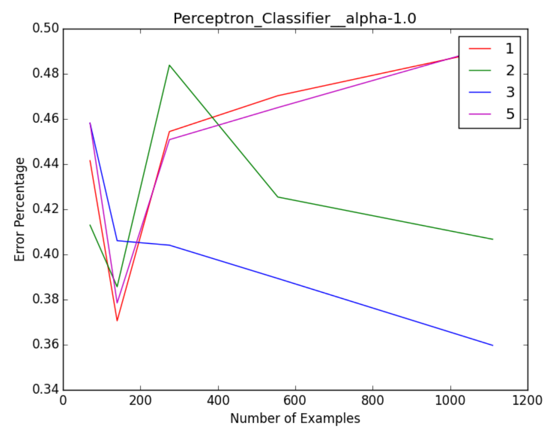
A Perceptron classifier was applied to the data with several fixed alpha values to determine the learning curves for various degrees of transformation for the feature set. The cross-validated learning curves for alpha values {0, 1e-6, 1e-3, and 1e0} are shown below:

Learning Curves





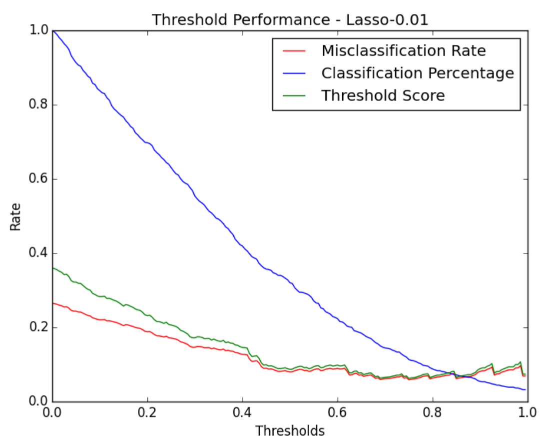


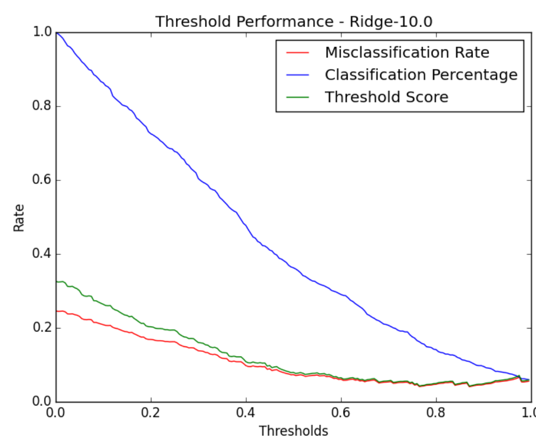


Once again we find that the results of the Perceptron classifier were significantly inferior to the Ridge and LASSO classifiers. Although there was some consistency in the results in the sense that degree 3 polynomials were performing the best, the overall performance of the classifier held around a 35%-40% misclassification rate – nearly a tenth more than LASSO or Ridge classifiers.

Performance and Threshold Scores

Ridge and Lasso performed the best with an average best-case training error of approximately 30%. This seems reasonable, since not all properties are expressly over-assessed or under assessed. We would like to see how well either of these classifiers perform when considering an "exclusion zone" - an area suitably close enough to the decision boundary that a correct classification cannot be guaranteed. The following plots show the total number of examples classified and the misclassification rate as a function of the threshold size. The 'threshold score' indicated below is essentially the number of wrong classifications divided by the number of right classifications, and should be as low as possible.





Conclusions

After testing several linear classifiers (Lasso, Ridge, Perceptron) and support-vector classifiers (Linear kernel, Quadratic kernel, and Radial-Basis Function (rbf) kernel). We found that Lasso and Ridge classifiers of degree 3 performed the best, with a generalized misclassification rate of less than 30%. Understanding that not all properties are strictly over or under assessed, we introduced a rejection region to the classifier to account for properties which are appropriately assessed. Using this rejection region, the classifier assigned approximately 50% of the properties an over/under rating with a reduced misclassification rate of 15%.

References

The machine learning library sci-kit learn was utilized to generate the classifiers for this project. Data was scraped from the web via the New York City Deparment of Finance. Both resources can be accessed via the following addresses:

1. Sci-Kit Learn: http://scikit-learn.org/stable/

2. New York Deparment of Finance: http://www1.nyc.gov/site/finance/index.page