House Prices: Advanced Regression Techniques

Nisha Chandwani[1], Sri Megha Vujjini[2]

Abstract

A House Price Index (HPI) measures the average price changes of houses and the methods generally used to calculate HPI are hedonistic regression, simple moving average and repeat sales regression. In this paper, we are aiming to predict the sale price of a residential home from Ames, Iowa using different regression techniques. The different models experimented with and applied to the dataset are Simple Linear Regression, Ridge Regression and Lasso Regression models. Initially, we apply the models locally and continue to calculate the predicted Sale Price by combining the models. The results show a significant difference in the error rate when we combine Lasso and Ridge regression models. With an error rate of 0.11838, we can summarize by saying that the model we developed is competitive and fares better than other regression models.

Keywords

Sale Price, Regression, Simple Linear, Lasso, Ridge.

[1] *Data Science, School of Informatics and Computing, Indiana University, Bloomington, IN, USA*
[2] *Data Science, School of Informatics and Computing, Indiana University, Bloomington, IN, USA*
***Corresponding author**: dalkilic@indiana.edu

Contents

Introduction

The recession in 2008 clearly demonstrates the importance of housing prices and interest rates on the economy. A good housing price model can predict the future housing prices and contribute to the establishment of real estate policies[1] but the exponential increase of the real estate market has made it difficult to predict the prices accurately and these unwelcome changes affect the individuals who are looking to buy or sell and the government as well.

Predicting the price of a house presents a unique set of challenges. The market is heterogeneous in both physical and geographical perspectives, which makes forecasting the price of a house difficult. Houses are individual, each with its own set of characteristics: the number of bathrooms, bedrooms, stories in structure, garage car spaces, square feet of finished living space, the presence of a private courtyard, a pool and/or hot-tub, the presence of an underground sprinkler system, the neighborhood and whether the house was new. The value of this bundle of characteristics is observed only when it is sold, which does not occur frequently. It is a time and energy consuming complex process involving visiting different websites and / or depending on different agents. The price varies in each structure and to avoid this, it is necessary to build a model which predicts the Sale Price of a house with most accuracy based on all the factors.

This is where data mining and machine learning come into play. Machine learning techniques are being used in multiple disciplines today because of their ability to learn and grow. We can use the concepts in real estate and instead of hard coded programs, we can build a model which learns from the dataset and teaches itself to give a better prediction. In this paper, we predict the selling price of residential properties in Ames, Iowa by using different regression models and aim to find the better model which gives the most accurate prediction.

1. Background

In this section, we introduce prediction analysis and give an overview of the housing market in Ames. We look at the dataset being used and some previous work done to predict the housing prices.

1.1 Prediction Analysis

Predictive analysis involves using a large dataset along with statistical algorithms, data mining and machine learning techniques to categorize the likelihood of future outcomes based on past data. Predictive models use approved results to develop (or train) a model that can be used on a new set of data. Modeling provides results in the form of predictions that representing the probability of target variable(s). There are two types of predictive models: Classification models and Regression Models.

Prediction analysis can be defined broadly as a three-step process.

1. Defining the problem
2. Obtaining the data
3. Building the model

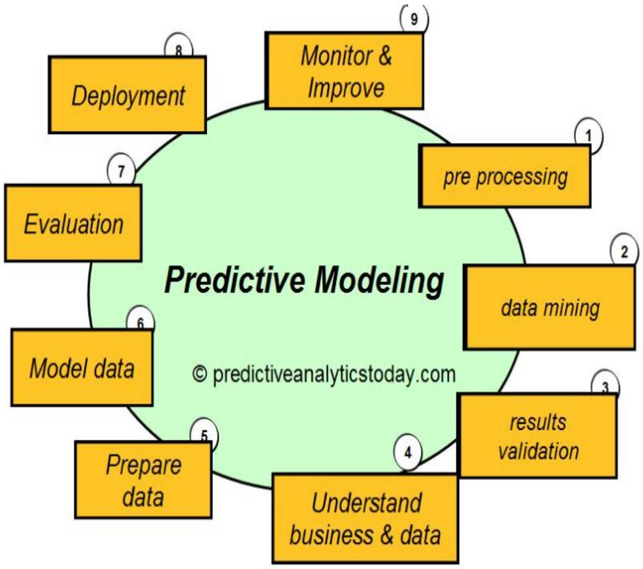

Fig 1: Predictive Analysis Model

In this paper, we build a regression model with the process steps as follows: problem can be defined as the housing price prediction, the data set being used is the housing data from Ames, Iowa and the models used are different types of regression.

1.2 Ames Housing Market

According to an article from Ames Tribune, the interest rates on the houses in Ames, Iowa have been increasing implying that its housing market continues to shrink while the prices of homes continue to rise. The market finds itself facing new challenges resulting from increased population growth and a lack of building space for new houses.

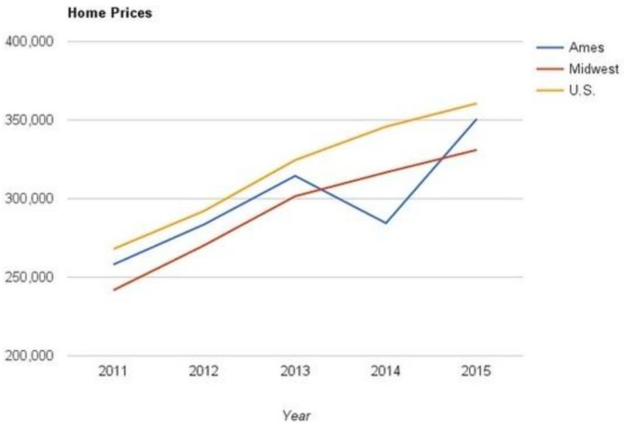

Fig 2: Housing Price Index of Ames, Iowa

Per Central Iowa Multiple Listing Service, in 2015, there were 648 homes sold in Ames and Ames' rural subdivisions. This number is up 6% from 613 in 2014. The number of houses sold in Ames has been rising since 2012, when the housing market started to recover from the recession. The average time that a home stayed on the market was down to 43 days in 2015 from 47 days in 2014. It can be concluded that the houses are being sold quickly due to a lack of properties available on the market.

Given how the market is tightening, it is extremely important to model a good relationship between the sale price and the various factors that affect it.

1.3 Related Work

Some of the existing models which predict the prices are the benchmark S&P/Case-Shiller model, which is a repeat sales model [2]. A modified form of the repeat sales models is used for the Home Price Index produced by the Office of Federal Housing Enterprise Oversight (OFHEO). Several criticisms have been made about repeat sales methods because the houses which have had some changes done between the two sales are removed from the analysis making the model useful only for a specific type of houses. Case and Quigley proposed a hybrid model that combined repeat sales methodology with hedonistic information so all sales could be included [3]; however, this requires housing characteristics that may be difficult to collect. There are some which use large scale models for house price predictions. DFM, FAVAR, LBVAR (spatial or non-spatial), Dynamic Stochastic General Equilibrium (DSGE) model, and forecast combination methods are the most popular methodologies for the analysis with large data sets. Gupta et al have discussed the differences in these models thoroughly [4]. In the paper 'Housing Value Forecasting Based on Machine Learning Methods', support vector machine (SVM), least squares support vector machine (LSSVM), and partial least squares (PLS) methods are used to forecast the home values. And these algorithms are compared per the predicted results. Experiments show that although the data set exhibits nonlinearity, SVM and LSSVM methods are superior to PLS [5] on dealing with the problem of nonlinearity. Based on the results from the above works and from our data set, advanced regression models are used in this paper.

1.4 Data Set

Dean De Cock from Truman State University compiled the Ames Housing dataset for use in data science education. The following information is from his paper, “Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project”. The data set contains 2930 observations and 79 explanatory data variables which describe almost every possible aspect of the residential homes in Ames, Iowa. Most of the variables are exactly the type of information that a typical home buyer would want to know about a potential property. For e.g., When was it built? How big is the lot? How many square feet of living space is in the dwelling?, etc. In general, the 20 continuous variables relate to various area dimensions for each observation. The 14 discrete variables typically quantify the number of items occurring within the house. There are many categorical variables (23 nominal, 23 ordinal) associated with this data set. They range from 2 to 28 classes.

Outliers: Although all known errors were corrected in the data, no observations have been removed due to unusual values and all final residential sales from the initial data set are included in the data. There are five observations that can be removed; three of them are true outliers and two of them are simply unusual sales.

2. Algorithm and Methodology

2.1 Overview

This section gives an outline of the regression technique in general and the different regression models used in the experiment. Once introduced, we specify the algorithm used to predict the sale price of the house.

2.2 Regression

For this problem, we are using the regression techniques for predicting sale price of the houses. A regression model can be used to predict the unknown variable using a set of independent variables. For this particular problem, we will be using multiple regression since we have more than one independent variables for predicting the dependent variable, Sale Price. For a regression model, the basic structure of data contains[6]

1. X: the matrix of input features
2. Y: the actual values of the dependent feature
3. Yhat : the predicted values of Y
4. W: weights or coefficients for each of the independent feature in X

Assuming, N to be the total number of data points and M as the total number of features, the predicted outcome for any data point, *i* is given by:

$$\hat{y}_i = \sum_{j=0}^{M} w_j * x_{ij}$$

i.e., the weighted sum of each data point with coefficients as the weights. The efficiency of the model depends on the weights; more optimum the weights, better the prediction model. The values for these weights are defined in multiple ways. These different ways are based on different criteria which distinguish the types of regression techniques.

2.2.1 Linear Regression

In simple linear regression, the objective function (also referred to as the cost function) that needs to be minimized, is the residual sum of squares (RSS), i.e., the sum of squared errors of the predicted outcome as compared to the actual outcome. Mathematically,

$$Cost(W) = RSS(W) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

To minimize this cost function, gradient descent algorithm which determines the gradient i.e., differential of the cost with respect to a weight, is used. The algorithm for linear regression on a high level is as follows:

1. Initialize weights
2. Repeat step 3 till not converged
3. Repeat steps 4 and 5 for all features (j = 0 to M)
4. Determine the gradient
5. Update the jth weight by subtracting, learning rate * gradient, from it:
   w(t+1) = w(t) – (learning rate * gradient)

In the above algorithm, convergence in step 2 refers to attaining the optimum solution which is checked based on the value of the gradient, in a predefined limit. If the gradient is small enough, we are very close to the optimum and further iteration won’t have much effect on the coefficients.

2.2.2 Ridge Regression

For ridge regression, the cost function to be minimized is RSS plus the sum of the squares of the magnitude of the weights (L2 regularization). Mathematically,

$$Cost(W) = RSS(W) + \\ \alpha * (\text{Sum of squares of weights})$$

The ridge regression generally works well in the presence of highly correlated features as it includes all of them in the model. It distributes the coefficients to each feature based on its correlation with the target variable. The major advantage of using ridge regression is that in shrinking the coefficient and reducing the model complexity. It prevents over-fitting on a major scale. However, we need to be careful before using ridge regression in the cases where there are large number of features (in millions) as it includes all the features in the model and can be computationally expensive.

2.2.3 Lasso Regression

For lasso regression, the cost function to be minimized is RSS plus the sum of absolute value of the magnitude of the weights (L1 regularization). Mathematically,

$$Cost(W) = RSS(W) +$$
$$\alpha * (\text{Sum of absolute value of weights})$$

Lasso is yet another regression technique that works well in case of highly correlated features which can also be used for feature selection because in a certain range, lasso coefficients become zero. The coefficients assigned by ridge regression are a reduced or penalized factor of the simple linear regression coefficients and are never zero. This makes lasso a more favorable choice for cases with millions of attributes since we can ignore the features with zero coefficients. It implies that feature selection is a part of lasso regression, thus providing computational advantage over ridge regression.

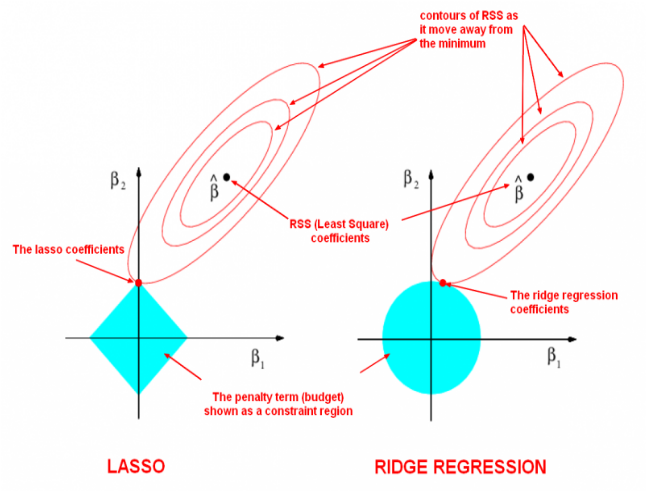
contours of RSS as it move away from the minimum
β_2
$\hat{\beta}$
RSS (Least Square) coefficients
The lasso coefficients
β_2
$\hat{\beta}$
The ridge regression coefficients
β_1
β_1
The penalty term (budget) shown as a constraint region
LASSO
RIDGE REGRESSION

Fig 3: Geometry of Lasso and Ridge Regression [7]

The above figure gives the depiction of the contours of the error and constraint functions for lasso and ridge regression. Weight coefficients are chosen to minimise both the the cost (contours) and the penalty (shaded area) terms. The best estimate for the model parameters (β) is at the point of intersection of the shaded area and the contours. However, the difference is that the Lasso penalty has sharper edges, leading the optimum estimates for β values to occur at the axes. Thus, smaller the number of β estimates produced, more the sparsity of the matrix of values meaning that some values of the co-efficients are exactly zero while others may be relatively large.

2.3 Algorithm

To summarize, ridge and lasso are powerful and regularized regression techniques that work well in presence of large correlated features. Since the data set contains multiple features for predicting the sale price, we can use these techniques without worrying about feature selection and collinearity between these features. The algorithm used can be concisely described as follows:

Step 1: Read the data from training and test data files.
Step 2: Combine the features (excluding target variable, sale price) from the training and test datasets for data cleaning and transformation.
Step 3: Handle missing values in the dataset obtained from step 2
Step 4: Split the dataset obtained from step 3 into numerical and categorical features
Step 5: Remove skewness from target variable using log+1 transformation
Step 6: Remove skewness from all the numerical features using log+1 transformation
Step 7: Scale all the numerical features
Step 8: Convert all the categorical variables to dummy variables
Step 9: Split the final pre-processed data back to training and test data sets
Step 10: Apply ridge and lasso regression techniques
Step 11: Assign weights to the results obtained from ridge and lasso and add these weighted values to predict the final sale price

3. Experiments and Results

The training set for housing data has 1460 rows and 79 explanatory features and the target feature, Sale Price. In this project, the model of predicting the sale price of a single-family house is affected by almost 79 variables which is a mix of nominal, ordinal and continuous variables. These 79 features consist of 44 categorical variables and 35 continuous variables.

3.1 Environment

- OS: Ubuntu 32-bit
- Python version: 2.3
- Python packages:
  - *sklearn* : matplotlib, pandas, numpy, preprocessing
  - *sklearn.linear_model* : Ridge, Lasso, LassoCV
- R version: 0.99.903

3.2 Data Cleaning

3.2.1 Handling Missing Values

We start our data cleaning by analyzing the missing values in the housing data set. It is not straight-forward for this dataset since we have multiple columns assigned as ‘NA’ to indicate that the feature is not relevant to that specific house. The value ‘NA’ for such features can be misleading, for e.g., there are houses which don’t have a basement but are assigned as ‘NA’. Thus, we need to first separate the data where NA represents a feature not present for a given house and the data where a house has a given feature but the dataset has NA indicating missing value for that feature.

For example, for the houses without a basement, we can replace ‘NA’ with ‘None’ to represent the absence of basement. However, we cannot do this staright away as it results in further complications. There are around 10 attributes related to basement and before interpreting NA as absence of basement, we need to make sure that no other basement related attribute has a value other than NA/0 for that particular house. On analyzing the given dataset, we find that the attribute, TotalBsmtSF, can be used as an indicator for basement. We found that any house that has this attribute set to 0 also has other basement related attributes set to either NA or 0. Thus, only if TotalBsmtSF for a given house is 0, we interpret it as ‘No Basement’ and thus only for these houses, we replace the NAs in the remaining basement related attributes with ‘None’ (categorical features) or 0 (continuous features). After we are done with the above step, the remaining NA values for the basement related attributes are cases where the given house has a basement but the dataset has a missing value for these attributes. In this way, we make sure that we handle the missing values for the ‘No Basement’ houses correctly.

We have applied similar approaches for other attributes as well where NA could imply that a specific feature is not relevant to that house. These features include-

- Alley – NA can be replaced by None
- FireplaceQu – NA can be replaced by None for houses where Fireplaces = 0 (indicating no fireplace)
- Fence - NA can be replaced by None
- MasVnrType - NA can be replaced by None for houses where MasVnrArea = 0
- Attributes related to Garage: NA can be replaced by None for these attributes where GarageArea = 0. However, there is one record in the dataset where GarageArea = NA. We can replace this value by 0.
- PoolQC - NA can be replaced by None for houses where PoolArea = 0
- MiscFeature- NA can be replaced by None for houses where MiscVal = 0

After replacing the NA values for the above features, where NA implied None or 0, we impute the remaining NAs which represent that a feature is present/applicable for the given house but the dataset has missing values for that feature. We do this by using the mice package in R and setting the parameter value for method = ‘cart’. We now have a dataset with no missing values for any of the explanatory features.

3.3 Data Transformation

After treating the missing values in the dataset, we start analyzing the data and exploring the relationship between the multiple features and the target attribute: Sale Price.

3.3.1 Removing Skewness

On analysis of the numeric features, we find that most of these are skewed, including the target variable(Sale Price). We thus apply a log+1 transformation to remove skewness from all numeric features and normalize the distribution of data. For instance, the below histograms show the difference in distribution of Sale Price, before and after the log+1 transformation.

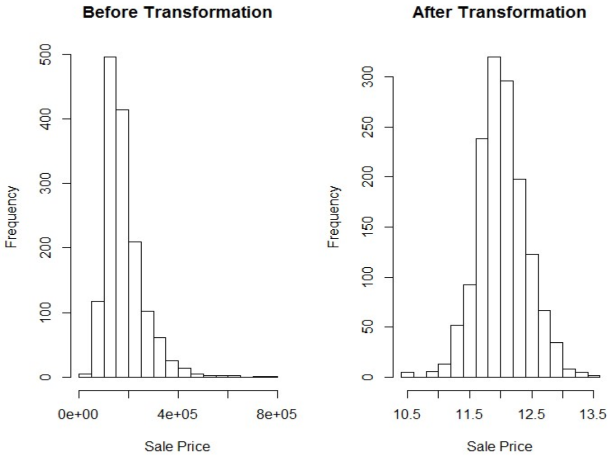

Fig 4: Sale Price before and after log+1 transformation

3.3.2 Scaling

Since we are using regularized linear regression techniques, scaling (or standardizing) the data is a good option. Scaling attributes will ensure they have a zero mean and a unit variance which makes the attributes comparable. In effect, scaling transforms the data to center it by removing the mean value of each attribute and then divides it by the standard deviation. Standardization of input data will allow the units of regression coefficients to be the same.

3.3.3 Exploratory Data Analysis

Continuous attributes:

We explore the relationship between the continuous attributes and the target attribute (Sale Price) as well as the relationship among these continuous attributes. We do this using the below correlation plot:

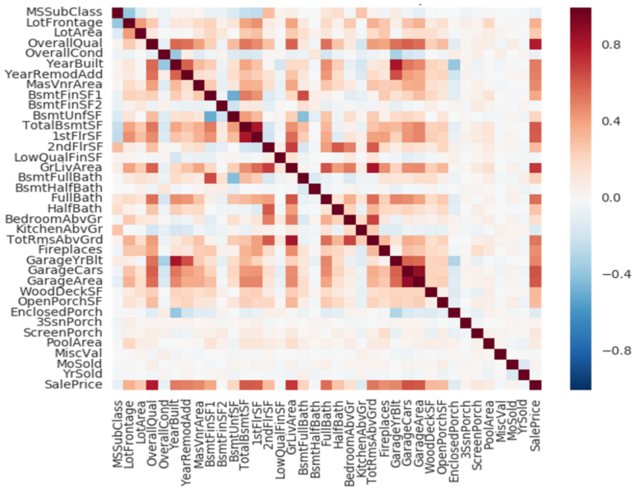

Fig 5: Correlation between continuous attributes

As seen in the plot, Sale Price has a strong correlation with many of the continuous attributes. For instance, there is a strong positive correlation between sale price and GrLivArea (ground living area square feet) and between sale price and OverallQual (overall material and finish quality) implying that, more the ground living area or better the overall quality of the house, higher the sale price. It also shows that there are very few continuous attributes with a negative correlation with the sale price. For example, kitchens above grade feature has a slightly negative correlation with the sale price, i.e., more the number of kitchens above grade, lower the sale price.

We also see that some of the attributes have strong correlation amongst themselves. For instance, GarageCars is strongly positively correlated with GarageArea and also OverallQual has a strong positive correlation with the YearBuilt.

This graph gives us a strong intuition of the attributes that will play a significant role in deciding the final sale price of a given house. Also, it shows strong correlation among some of the attributes(collinearity), which implies that linear regression might not produce good results and rather we should use regularization regression techniques to handle this collinearity in the input data features.

Categorical Attributes:
To explore the relationship between the categorical attributes and the sale price, we used box plots. For instance, lets see the impact of neighborhood on sale price.

The plot at the end of this section shows that Brook Side and South & West of Iowa State University have houses with lower sales price while North ridge and North ridge Heights are expensive neighborhoods (though we do see some outliers for these). Thus, neighborhood is an important categorical attribute in deciding the sale price of houses.

Similarly, we plot the relationship between all the other categorical attributes and the target attribute (Sale Price) and assess the importance of these features in the sale price evaluation.

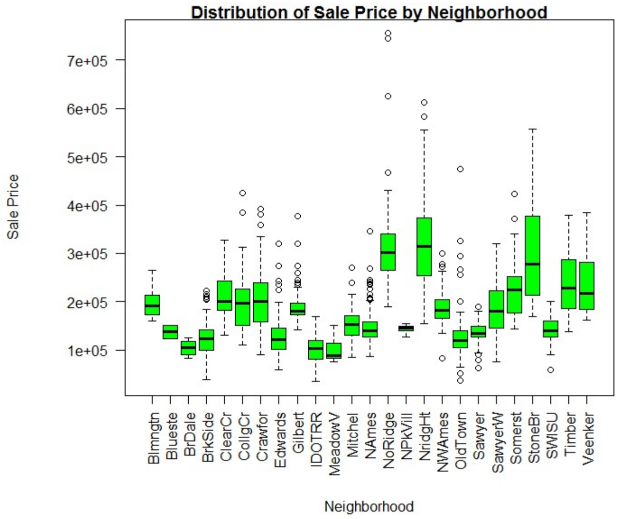

Fig 6: Relationship between categorical attribute, Neighborhood and Sale Price

3.4 Data Modeling

Once we are done with the data pre-processing, we can experiment by applying various advanced regression techniques to our dataset for predicting the sale price of the houses.

Since we have collinearity between the attributes, we will apply regularized regression techniques such as Lasso and Ridge regression. Also, by using these techniques, we do not need to worry about feature selection since they assign weights to each feature based on its significance in predicting the sale price.

One thing to note is that, for both ridge and lasso regression, we need to define the value of the tuning parameter α. This is done using cross validation which checks for a range of values and selects the one which gives the lowest prediction error. We found $\alpha = 0.00048$ giving the lowest negative mean square error for lasso and $\alpha = 15$ giving the lowest error for ridge regression. Same can be inferred from the plots given below:

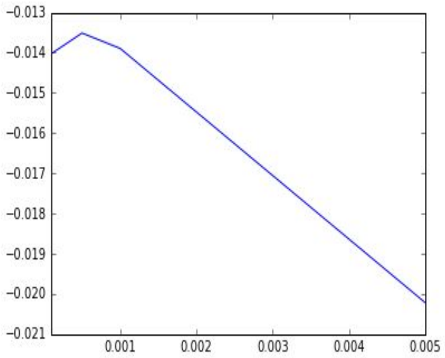

Fig 7: Effect of α on error value in Lasso Regression

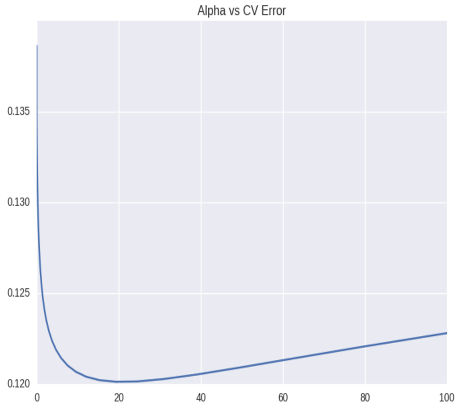

Fig 8: Effect of α on error value in Ridge Regression

We first start with Lasso Regression, a regularized regression technique which adds penalty equivalent to the absolute value of the magnitude of the coefficients, i.e., L1 regularization.

3.5 Results

On applying Lasso regression on this pre-processed dataset, the Root-Mean-Squared-Error (RMSE) between the predicted value and the actual sale price is found to be 0.12185.
On applying Ridge regression on this pre-processed dataset, the RMSE is 0.11871. Thus, in this case, ridge regression performed better than lasso regression by a good margin.

We now experiment using both the regression techniques together. We combine the outputs from both ridge and lasso regression such that we assign 70% weightage to the results from ridge and 30% to the results from lasso. We tried different combinations of weights for these two techniques and found the above which resulted in the lowest RMSE of 0.11838.

The results can be summarized as follows:

Regression	RMSE
Lasso	0.12185
Ridge	0.11871
0.5*Lasso + 0.5*Ridge	0.11840
0.3*Lasso + 0.7*Ridge	0.11838

4. Summary and Conclusions

From the results, we see that the regression model employed generated gives predictions for the sale price which are quite accurately close to the actual sale price. We can thus conclude that, after pre-processing the data such that it satisfies the assumptions of regression, a combination of lasso and ridge delivers good results for sale price prediction. However, as part of future work, we can try incorporating few more techniques in the existing model, such as random forest regression for example, and predict sale price by taking votes from all these models. Also, with such large number of features available for predicting the sale price, we could try some feature engineering to make the algorithms work better. This would require gaining domain knowledge in depth to create new features from the already existing ones.

Acknowledgments

Thanks to Dean De Cock for compiling the Ames Housing dataset which helped in understanding the dataset for housing project.
Thanks to Alexandru Papiu for the great Kaggle kernel on regularized linear model.
Special thanks to our professor, Mehmet Dalkilic and the assistant instructors (AIs) for clarifying our doubts and helping us understand the requirement of this project.

References

[1] AnHai Doan, Pedro Domingos, and Alon Y Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *ACM Sigmod Record*, volume 30, pages 509–520. ACM, 2001.

[2] Byeonghwa Park and Jae Kwon Bae. Using machine learning algorithms for housing price prediction: The case of fairfax county, virginia housing data. *Expert Systems with Applications*, 42(6):2928 – 2934, 2015.

[3] Bradford Case, Henry O Pollakowski, and Susan M Wachter. On choosing among house price index methodologies. *Real estate economics*, 19(3):286–307, 1991.

[4] Rangan Gupta and Alain Kabundi. Forecasting macroeconomic variables using large datasets: Dynamic factor model versus large-scale bvars. *Indian Economic Review*, pages 23–40, 2011.

[5] Gongzhu Hu, Jinping Wang, and Wenying Feng. Multivariate regression modeling for home value estimates with evaluation using maximum information coefficient. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2012*, pages 69–81. Springer, 2013.

[6] Aaryshay Jain. A Complete Tutorial on Ridge and Lasso Regression in Python. https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso\-regression-python/, 2016. [Online; accessed 14-Dec-2016].

[7] Niall Martin. Shrinkage Methods: Ridge Vs. Lasso Regression. https://niallmartin.me/2016/05/12/shrinkage-methods-ridge\-and-lasso-regression/, 2016. [Online; accessed 14-Dec-2016].