Group 4 - Housing Price Prediction

Erik Alvstad, Lance Halsted, Ruiyi He, Ashley Li, Abdullah Mohammed
William Schmidt, Qilin Wang, Cai Xu, Ruyi Yang

July 26, 2021

Introduction

Github Link

The housing market contributes between 15-18% of the United States GDP, depending on the year (NAHB). Such a relatively high contribution, coupled with the necessity of such products makes datasets composed of explanatory variables (*id est* size, number of bedrooms, number of bathrooms, et cetera) and their sale price, both easy to obtain and quite substantial. Dean De Cock at Truman State University collected 2930 real estate sale observations, from the city of Ames, Iowa, with 80 explanatory variables (De Cock). The dataset has since been used as an educational tool for undergraduate regression courses. With this in mind, our team has set out to create a machine learning algorithm to accurately predict the sale price of a house from these explanatory variables based on this dataset.

Background

The basic premise of machine learning is to produce a model that accurately predicts relationships in data from a statistical sample of a population. Producing a model that does not fully capture the relationships in the sample such that the model cannot predict those of the population well is referred to as underfitting. Producing a model that predicts the sample well by describing "noise" in the data, rather than describing overall trends, and therefore, cannot accurately predict those in the population is referred to as overfitting. The goal of machine learning, therefore, is to produce a model that is detailed enough to prevent underfitting, while also reducing the detail such that overfitting does not occur.

Linear regression analysis is a statistical method of determining the linear relationship between two variables. The most basic of these is simple linear regression, wherein constants β_0 and β_1 are selected such that some error function of the difference between $y = \beta_0 + \beta_1 x$ and the actual values of y is minimized. Multiple linear regression is a methodology by which the same procedures are performed, however, several explanatory variables (x_1, x_2, etc) are analyzed: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$ (Pedregosa *et al.*).

There are several methods for determining the constants that reduce the error to its global minimum. While the absolute global minimum can be calculated with linear algebra, with large datasets this becomes unfeasible and gradient descent methodologies become preferential. Among the most common methodologies in the study of machine learning include ordinary least-squares regression, stochastic gradient descent regression, ridge-regularized regression, lasso-regularized regression, support vector regression, and polynomial regression.

Gradient descent methodologies, follow a simple procedure. First, random values for each constant are chosen. Then, expected values are calculated for some portion of the dataset. The error function's derivative is calculated, and the constants are changed to move the error function down its gradient (Pedregosa *et al.*). This is repeated until a local minimum is found.

Ordinary least-squares (OLS) regression is an analytical methodology of determining the global optimum for a linear regression. It uses matrix multiplication, and, therefore, suffers from a large time complexity (Pedregosa *et al.*).

Stochastic gradient descent (SGD) is a methodology that seeks to improve the speed of OLS regression. SGD uses a numerical approach compared to the analytical approach used in OLS. SGD uses a learning rate hyperparameter, a, to determine the size of steps in moving toward the minimum: $w_j = w_j - \alpha \frac{\delta RSS}{\delta w_j}$. However, SGD is not guaranteed to give an optimal solution since it may find a local minimum instead of the global minimum.

Polynomial regression is a special case of linear regression where an nth-degree polynomial is used to model the relationship between the n independent variables and the target variable (Sharma). The model function is linear with respect to the model weights. The formula, therefore, is $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + ... + \beta_n x^n$. Higher-order polynomial regression can lead to overfitting of the training data.

Ridge-regularized regression is a methodology that seeks to prevent overfitting by minimizing the values of the constants. This is achieved by appending the sum of squares of the regression constants to the error function, multiplied by an additional hyperparameter (Pedregosa *et al.*).

Lasso-regularized regression is a methodology almost identical to ridge-regularized regression, however, instead of appending the sum of squares of the regression constants, the sum of the regression constants is appended (Pedregosa *et al.*).

Support Vector Regression (SVR) uses the same principle as SVM, but applies it to regression problems instead of classification problems. In SVR, the best fit line is a hyperplane that has a maximum number of points within a threshold distance. The SVR kernel can be either linear, polynomial or nonlinear. Due to SVR's fit-time complexity, it is not suitable for very large datasets (Raj).

Literature Review

The prediction of sale price has numerous applications in real-world service industries. Online shopping services may want to determine the best price for a given individual shopper, and investors may want to predict if and when a stock price may rise and fall. The housing market is a market in which the features of the product for sale vary wildly, and price has many explanatory variables. As such, it is a prime application of machine learning algorithms. It is almost certain that linear regression analyses provide effective means of determining the sale price of the house, though they are often used in conjunction with other methodologies.

As a problem that requires the use of supervised learning, regression analyses are often used for their prediction. In Truong *et al.'s* analysis of housing prices in Beijing, it was found that a regression analysis of the sale price with features as explanatory variables was satisfactory with a large sample of 300,000 observations. Similarly, in an analysis of housing prices in Melbourne City, Australia, students at Macquarie University used linear and polynomial regression to produce a satisfactory model using a data set of about 35,000 samples with 21 explanatory variables. (The Danh Phan).

Ensemble machine learning techniques are often found to be best for this problem. In Varma *et al.'s* analysis of machine learning algorithms, it was found that an optimal machine learning methodology for house sale prices in Mumbai, India, was an ensemble methodology, featuring regression and neural networks. Ensemble methodologies have been found to perform better than pure regressions (Varma *et al.*).

In Yichen Zhou's masters thesis, a linear regression as well as a lasso regularized linear regression are performed on the De Cock dataset and tested for accuracy. OLS regression analysis was found to perform better with this dataset than Lasso-regularized least squares regression (Zhou).

0.1 Dataset

Our team obtained a subset of the original dataset from Kaggle. Kaggle is a public dataset repository. The dataset involved had 1460 instances, each with 80 different attributes. Of the attributes, 28 were numeric, and 52 were categorical.

The dataset's attributes were titled with strings with 12 characters or less. For the sake of reducing confusion, the attributes and their meanings are summarized in the *Attributes* section below.

0.2 Attributes

1. ID: which instance was recorded (irr)
2. MSSubClass: the building class (cat)
3. MSZoning: the zoning classification (cat)
4. LotFrontage: the length of street connected to the perimeter (cat)
5. LotArea: the lot size in square feet (num)
6. Street: the type of road access (cat)
7. Alley: the type of alley access (cat)
8. LotShape: the lot shape (cat)
9. LandContour: flatness (cat)
10. Utilities: the type of utilities available (cat)
11. LotConfig: the lot configuration (cat)
12. LandSlope: the slope of the property (cat)
13. Neighborhood: the location (cat)
14. Condition1: the proximity to various railroads and road features (cat)
15. Condition2: the proximity to various railroads and road features (cat)
16. BldgType: the type of structure (cat)
17. HouseStyle: the style of the structure (cat)
18. OverallQual: the quality of the material and finish (cat)
19. OverallCond: the condition (cat)
20. YearBuilt: the original construction date (cat)
21. YearRemodAdd: the date of any remodeling (irr)
22. RoofStyle: the style of roofing (cat)
23. RoofMatl: the roofing material (cat)
24. Exterior1st: the first exterior covering (cat)
25. Exterior2nd: the first exterior covering (cat)
26. MasVnrType: the masonry veneer (cat)
27. MasVnrArea: the masonry veneer area in square feet (num)
28. ExterQual: the exterior quality (cat)
29. ExterCond: the exterior condition (cat)
30. Foundation: the type of foundation (cat)
31. BsmtQual: the height of the basement (cat)
32. BsmtCond: the condition of the basement (cat)
33. BsmtExposure: the exposure of the basement walkout (cat)
34. BsmtFinType1: the rating of the first basement finish (cat)
35. BsmtFinSF1: the area of the first basement finish (num)
36. BsmtFinType2: the rating of the second basement finish (cat)

37. BsmtFinSF2: the area of the second basement finish (num)
38. BsmtUnfSF: the square feet of the unfinished basement area (num)
39. TotalBsmtSF: the total square feet of basement area (num)
40. Heating: the type of heating (cat)
41. HeatingQC: the heating quality and condition (cat)
42. CentralAir: whether there was central air conditioning on the property (cat)
43. Electrical: the type of electrical system (cat)
44. 1stFlrSF: the area of the first floor in square feet (num)
45. 2ndFlrSF: the area of the second floor in square feet (num)
46. LowQualFinSF: the area of low quality finish in square feet (num)
47. GrLivArea: the living area above ground in square feet (num)
48. BsmtFullBath: the amount of full bathrooms in the basement (num)
49. BsmtHalfBath: the amount of half bathrooms in the basement (num)
50. FullBath: the amount of full bathrooms above ground (num)
51. HalfBath: the amount of half bathrooms above ground (num)
52. Bedroom: the amount of bedrooms above ground (num)
53. Kitchen: the amount of kitchens above ground (num)
54. KitchenQual: the quality of the kitchens (cat)
55. TotRmsAbvGrd: the total amount of non-bathroom rooms above ground (num)
56. Functional: the home functionality (cat)
57. Fireplaces: the amount of fireplaces (num)
58. FireplaceQu: the quality of the fireplaces (cat)
59. GarageType: the location of the garage (cat)
60. GarageYrBlt: the year built of the garage (irr)
61. GarageFinish: the type of interior finish of the garage (cat)
62. GarageCars: the size of the garage in car capacity (num)
63. GarageArea: the size of the garage in square feet (num)
64. GarageQual: the quality of the garage (cat)
65. GarageCond: the condition of the garage (cat)
66. PavedDrive: the amount of driveway paved (cat)
67. WoodDeckSF: the area of wood deck in square feet (num)
68. OpenPorchSF: the area of open porch in square feet (num)
69. EnclosedPorch: the area of enclosed porch in square feet (num)
70. 3SsnPorch: the area of three season porch in square feet (num)
71. ScreenPorch: the area of screen porch in square feet (num)
72. PoolArea: the area of pool (num)
73. PoolQC: the quality of the pool (cat)
74. Fence: the quality of the fence (cat)
75. MiscFeature: miscellaneous features unlisted elsewhere (cat)
76. MiscVal: the value of the miscellaneous feature (num)
77. MoSold: the month sold (cat)
78. YrSold: the year sold (cat)
79. SaleType: the type of sale (cat)
80. SaleCondition: the condition of the sale (cat)

Analysis

The attributes listed were not statistically independent. Several simple linear regressions performed between attributes yielded correlation values far above 0.5. The largest of which was between *GarageArea* and *GarageCars*, which had a correlation of 0.88. Some other notable attribute correlations include *GarageYrBlt* and *YearBuilt*, with a correlation of 0.83, in addition to *1stFlrSF* and *TotalBsmtSF*, with a correlation of 0.82. Indeed, the largest of the correlations usually emerged due to the attributes in question having similar, if not identical meanings (*GarageArea* and *GarageCars* both state how large the garage of the instance is, albeit with different units). There are few, if any, attributes that have nonzero correlations, and most fall around 0.20.

The attributes were, for the most part, normally distributed. The distributions of the numeric attributes do not deviate far from a normal distribution. With how extensive the numeric attributes were, and the target variable itself being numeric, the dataset in question was an adequate candidate for regression analysis.

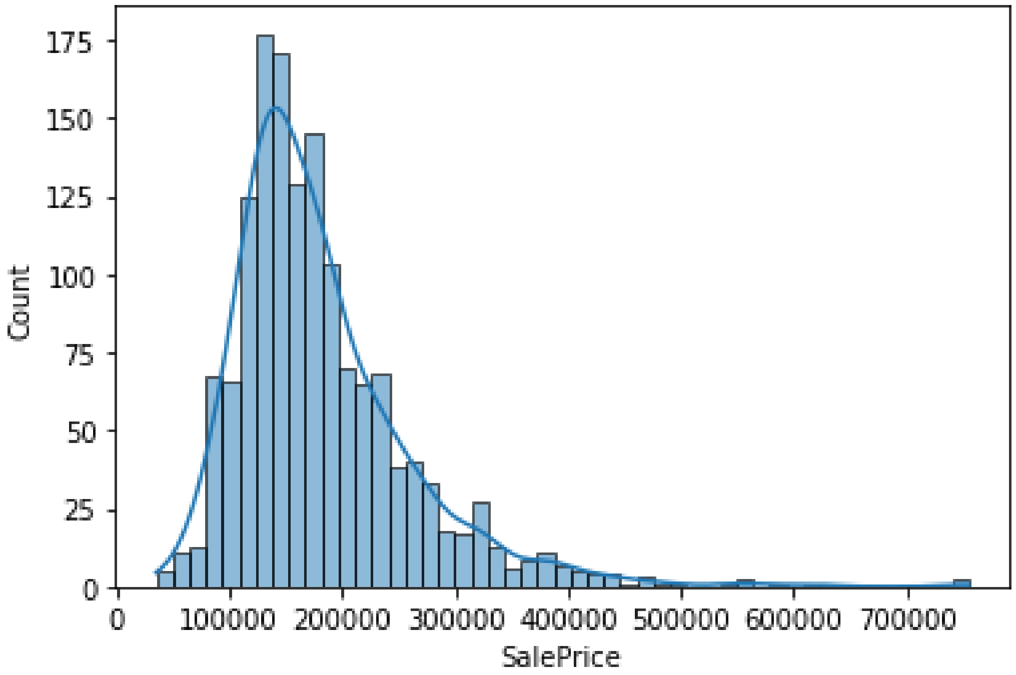

Figure 1.1: A histogram depicting the distribution of the *SalePrice attribute*

The target variable, *SalePrice*, had a right-skewed normal distribution. As seen in Figure 1.1, there is a single peak, and few, if any, irregularities in the data.

Methodology

Our team preprocessed the dataset and trained and tested the model using the Jupyter Notebook running Python 3. The libraries used included Numpy, Pandas, and SciKit-Learn. Numpy is an open-source Python library for numeric computation that was created in 2005. Pandas is an open-source Python library for data analysis that was created in 2008. SciKit-Learn is a Python library for machine learning that was created in 2007.

0.3 Dataset Preprocessing

The provided dataset underwent several transformations before it was used for model training. Heuristics were used to determine initial preprocessing steps, then the methodology was refined through experimentation.

The dataset was first converted into a Pandas dataframe. Next, the target variable was separated from the training variables. Then, each training variable was grouped into one of three categories: categorical variables, numeric variables, and irrelevant variables - listed as cat, num, and irr in the *Attributes* section.

Categorical variables were variables in which two or more non-ordinal categories were possible for each instance in the dataframe. Such variables were encoded via one-hot encoding. For some categorical variables, "NA" values were supported and indicated that the feature in question was not present. For such variables, "NA" entries were converted to the string "None" to avoid Pandas treating them as missing values. For all remaining categorical features, "NA" values were processed as missing entries. Furthermore, the numeric variables *LotFrontage* and *MasVnrArea* used "NA" to denote a value of 0. In these cases, all "NA" values were converted to 0 to avoid Pandas treating them as missing.

Numeric variables were normalized to their respective z-scores, following a simple normal distribution. Outliers were systematically removed by dropping any instance of a numeric variable falling 3.5 standard deviations or greater from the mean. This threshold was determined through experimentally observing the effect it had on model performance. Both removing outliers and preserving the robustness of the model were the two main factors taken into account.

Variables that were deemed irrelevant were those that did not add useful information to the model - such as the ID of the house - or highly correlated variables. Correlation was observed using the 'pandas.DataFrame.corr' method. Irrelevant features were simply dropped from the dataframe.

Once each of the explanatory variables had been effectively transformed, the categorical, numeric, and target data frames were concatenated into a single object. From here, a .csv file was saved locally that would be used for model training.

0.4 Model Training

A total of six different models were trained and subsequently tested. An OLS regression, a SGD linear regression, a ridge-regularized linear regression, a lasso-regularized linear regression, a support vector regression, and a polynomial (quadratic) regression were each performed on the dataset. Each model was performed with grid-search hyperparameter tuning and was subject to 10-fold cross validation.

0.5 Model Testing

Each model was tested for its root-mean-squared error on the respective testing sets of the 10-fold cross validation, with the averages calculated and compared. The model with the smallest average root-mean-squared error was selected as the final model.

Results

	OLSR	RidgeR	LassoR	SGDR	SVR	PR
R2	-2.400	0.910	0.913	0.901	0.912	0.896
RMSE	3.98e15	2.04e4	2.00e4	2.14e4	2.03e4	2.15e4
Iter		5	751	50		
Alpha		12	65	1.8e-3		
FitI	False					
Norm	True					
Pos	False					

Figure 2.1: The average r-squared values and root-mean-squared errors of each optimized model, as well as the optimal hyperparameters: Iter is the number of iterations performed; Alpha is the alpha value for gradient descent; FitI fits the model to the intercept; Norm normalizes the data; Pos makes all the coefficients positive; empty cells mean that the hyperparameter is N/A.

After all of the procedures were performed, each of our models was tested, and their outputs are depicted in Figure 2.1.

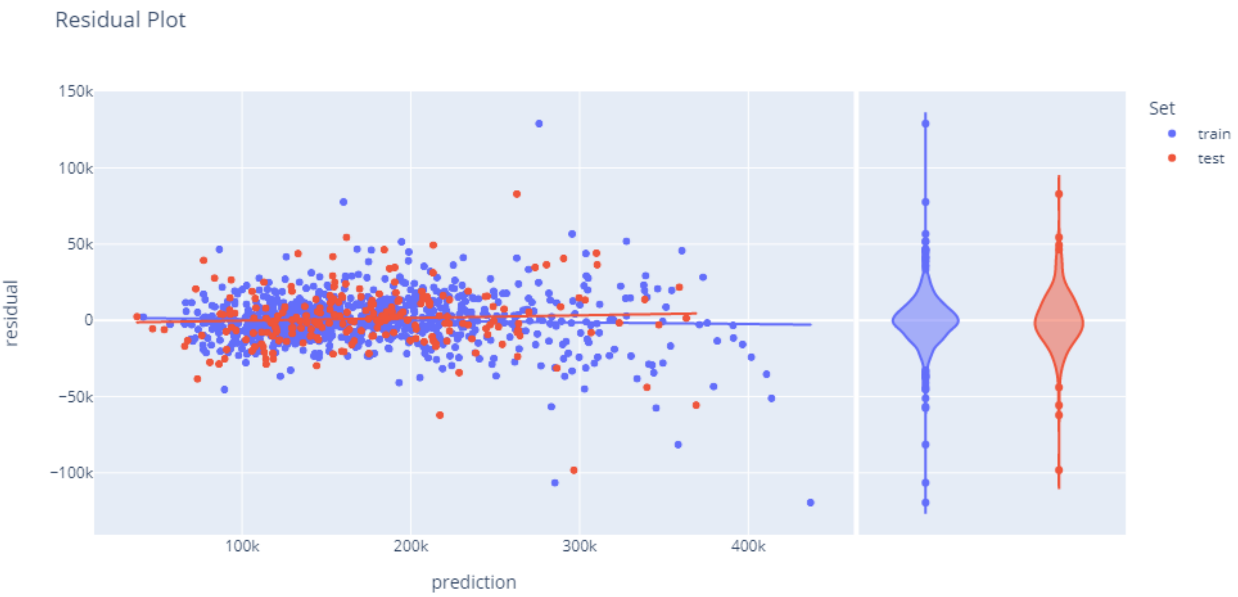

Figure 2.2: Residual plot for Lasso-regularized regression model

Since Lasso performed best, a residual plot of the predicted price vs the actual price in the test and training sets is depicted in Figure 2.2. Ideally, the residuals should be independent of the target variable and normally distributed. The Lasso residual plot has fairly good symmetry around the residual=0 axis, especially for houses that cost below $350k.

Discussion

While most of the regression analyses performed had similar results, one of the major outliers was the OLS regression. It performed the worst, by far, with a root mean squared error of 3.98e15. While it is hard to say why this regression failed to produce the desired outcome, several hypotheses are possible. Perhaps the relationships that exist in the dataset are not strictly linear, and a linear regression is not applicable. Indeed, the polynomial regression, which, in this experiment, performed a quadratic regression analysis, did perform better than the OLS regression. Perhaps the optimal linear regression has very large coefficients that do not fit the population well. Ridge regularized linear regression does perform better than OLS regression, and ridge regularization is a methodology to prevent large coefficients in the model.

The polynomial regression performed quite well; 2.15e4 was a substantially lower RMSE than the OLS RMSE. Perhaps this is due to some of the relationships in the sample being quadratic rather than strictly linear. This may be the case, however, the polynomial regression was not best, and so, there may have been other problems with this regression that went unaddressed.

The stochastic gradient descent model performed well in this experiment. With a RMSE of 2.14e4, this model performed better than the OLS. Since the OLS is likely a global minimum, this would seem to mean that SGD found a local minimum with better performance than the OLS global minimum. As stated prior, this may be because the SGD had smaller coefficients relative to the OLS model.

The ridge regularization regression model performed better than the stochastic gradient descent model. Since the ridge regression is mainly used to prevent large coefficients from reducing the testing accuracy of regression models, the ridge regression's success, relative to both the SGD and the OLS indicate that this factor is relevant to the dataset.

The SVR model performed very well. Ultimately, this may be due to its design; the methodology of determining the optimal hyperplane was different from the other regression methodologies. However, it was not the best.

The Lasso-regularization model performed best. With a r-squared value of 0.913 and a RMSE of 2.00e4, this function was substantially better than the SGD and the OLS, indicating that, at least for linear regressions, performing regularization functions to reduce coefficients is an optimal way of dealing with the sample size being as small as the De Cock dataset's.

Conclusion

The dataset that all of the models discussed were trained with had several significant problems: it was small and had several, highly-correlated attributes. These characteristics are all factors that increase the likelihood of large coefficients. The best model was the lasso-regularized linear regression, which was produced with a methodology that seeks to reduce the size of coefficients drastically. It is very likely that this was a major problem that each of the models faced. The OLS regression model performed the worst, and its regression is likely the global minimum of the training set.

With this dataset and with others, experts in the field of machine learning use linear regression techniques with neural networks and other ensemble models to determine relationships in housing prices (Varma *et al.*). While these models are beyond the scope of this paper, it has been effectively shown that linear regression can be used as a single method for discovering these relationships.

While large datasets require vast computational power to produce optimal models, smaller datasets induce problems with overfitting that can only be overcome with sophisticated methodologies and hyperparameter selection. Collecting larger datasets remains the most effective way of improving the accuracy of machine learning models.

References

A. Varma, A. Sarma, S. Doshi and R. Nair, ”House Price Prediction Using Machine Learning and Neural Networks,” *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 1936-1939, doi: 10.1109/ICICCT.2018.8473231.

Dean De Cock. “Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project” *Journal of Statistics Education*, Volume 19, Number 3. 2011.

“House Prices - Advanced Regression Techniques.” *Kaggle*, www.kaggle.com/c/house-prices-advanced-regression-techniques/overview.

“Housing’s Contribution to Gross Domestic Product.” NAHB, www.nahb.org/News-and-Economics/Housing-Economics/Housings-Economic-Impact/Housings-Contribution-to-Gross-Domestic-Product.

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Sharma, Abhishek, Introduction to Polynomial Regression (with Python Implementation), Retrieved from https://www.analyticsvidhya.com/blog/2020/03/polynomial-regression-python/

Raj, Ashwin, Unlocking the True Power of Support Vector Regression - Using Support Vector Machine for Regression Problems. Retrieved from https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0

The Danh Phan, Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia, Retrieved from https://ieeexplore.ieee.org/abstract/document/8614000

Truong, Quang, *et al.* “Housing Price Prediction via Improved Machine Learning Techniques.” Procedia Computer Science, vol. 174, Elsevier BV, 2020, pp. 433–442. Crossref, doi:10.1016/j.procs.

Zhou, Y. (2020). Housing Sale Price Prediction Using Machine Learning Algorithms. UCLA. ProQuest ID: Zhou_ucla_0031N_18606. Merritt ID: ark:/13030/m5wh7xdt. Retrieved from https://escholarship.org/uc/item/3ft2m7z5