Housing Prices

Ridge vs Lasso regressions for the prediction of the median house price

Anna Olena Zhab'yak, Michele Maione

*Departments of Economics and Quantitative Methods &Informatics*

*Faculties of Data science and Economics & Informatics*

*University of Milan*

**Summary** – This paper is trying to answer to the question what the best regression model is to predict the median house price.

The hedonic theory is exploited and models as the Ridge and the Lasso regression are used on a cross sectional dataset of housing prices in order to predict the median house value. Applying the cross-validation we estimate the parameters and then evaluate the model. Then PCA is implemented in order to improve the risk estimator. In section 1 we introduce the problem of housing prices in the U.S. and the used approach in this work. Section 2 is dedicated to the literature about regressions in predicting the price of real estates and 3rd section the notation is clarified in order to simplify the understanding of the notions. Our experiment is described in section 4 with consequential critical comments and evaluations.

# Introduction and description of the problem

The hedonic theory identifies the attributes as implicitly embodied in goods and their observable market prices, so extending this concept to the housing prices we can see the attributes as the house’s characteristics that are determinant for the final value. It exploits the consumer theory and her willingness to pay depending on the utility gained from the bundle of aggregated attributes. Each attribute differently influences the price and its strength is given by the estimated coefficient.

Our work starts from a real problem of housing prices in the United States, where the economical purposes and the low mortgage rates incentive a solid and hot real estate market*.*[[1]](#footnote-1)Indeed, the U.S. is one of the most stable and secure countries for real estate investment in the recent years[[2]](#footnote-2). It is estimated that household wealth is nearly 50% invested in real estate and the owner-occupied housing rate in July 2019was about 63,5%[[3]](#footnote-3). However, the U.S. real estate market was not always as reliable as today, indeed the sudden bubble of the housing market of 2006-2007 preceding the Great Recession and its subsequent burs is clear evidence of the system weaknesses. The speculation on the housing prices and their extremely high values is due to the lack of information caused by the manipulations of major players in the real estate sector[[4]](#footnote-4). For these reasons, the task of predicting the value of a house becomes crucial, as the constructed house price model can influence the economic growth and improve the efficiency of the real estate market. An accurate prediction model is significant and helps to fill up an information gap for the prospective homeowners, policy makers and other real estate market participants, such as, mortgage lenders and insurers[[5]](#footnote-5). Modelling house prices presents some issues, for example the median value might be extremely influenced by the value of the sold properties in the area with similar characteristics[[6]](#footnote-6) or the prediction could become wrong due to exogenous factors influencing the prices. Indeed, the economic health reflects in the market according to the supply and demand law so any shock will affect the current prices. Moreover, working on a large dataset, like the one used in this work, can lead to the so-called multicollinearity of the features which tend to overfit when it comes to implement the algorithm predicting the value.

The classic OLS regression has the desired property of being unbiased, but it can suffer of overfitting and have a huge variance in those cases where features are highly correlated. To pull down the variance and obtain more biased estimator a regularization technique is necessary*.* The focus of this paper is therefore on two regularization techniques, the Ridge and Lasso regression. The Ridge regression[[7]](#footnote-7) is a useful tool for improving prediction in regression tasks with highly correlated predictors[[8]](#footnote-8). Lasso regression is also used to handle high dimensional databases where the features are correlated, and this technique shrinks some of them to zero, performing a feature selection with a consequent dimension reduction. Both methods act on the coefficients by introducing a penalty on them in order to make more effort to the most informative ones, this way minimizing overfitting of the data and solving the multicollinearity problem. The impact of each attribute on the predicted price is given by the value of the coefficient, higher coefficients mean higher influence. The penalty is the tool through which we perform the regularization, also called tunning parameter, it controls the bias-variance trade-off and the selection of it is crucial. For choosing the regularization parameter in practice, cross-validation (CV) is widely used.

# Most important related works

Many works have been developed to predict the median house value with models of different complexity [see Manjula et all, 2017].

The concept of hedonic prices was developed by Rosen (1974), however the first the first implementing the hedonic model to the house sector was Lancaster (1966). Griliches (1971)[[9]](#footnote-9) provided the reading of a commodity, such as a house, as an aggregation of individual components or attributes. Timothy Oladunni & Sharad Sharma (2016) and Limsombunchai et al. (2004) have showed that the price of a property is predictable exploiting the hedonic theory, comparing the hedonic regression in comparison with other algorithms[[10]](#footnote-10).

Dubin (1998) has developed a work to predicted house prices using MLS data, even though exploiting different algorithms for the prediction, such as kriging algorithm to create an accurate spatial interpolation of house prices. Other as Xin and Khalid (2018) have used ridge and lasso regression to deal with multicollinearity of features on a time series database for predicting the housing price.

Hoerl and Kennard (1970) firstly introduced the Ridge regression as biased estimator for nonorthogonal problems.

The asymptotic properties of ridge have been widely studied (e.g., Tulino & Verdu´, 2004; Serdobolskii, 2007; Couillet & Debbah, 2011; Dicker, 2016; Dobriban & Wager, 2018, etc).

For the validation approach we refer to the cross validation. which biased estimation of the error is known (Hastie et al., 2009, p. 243), since it uses a smaller amount of data than the entire dataset.[[11]](#footnote-11) However, we can apply a bias-control , see Liu and Dobriban (2020)[[12]](#footnote-12) , for example via k-fold cross validation, see Ray (2018), since there is an inverse relation between the k size and bias, if the first grows the latter goes down.

furthermore – inoltre

# Notation and relevant definitions (review)

The goal of the regression is to generate a prediction such that the loss function is small for most data points , where is a predictor from the labels set with the regularization term and data domain; the prediction mistakes are a function of the difference.

## Hedonic model

The price can be written as follows

In this case the price is a function of:

* longitude,
* latitude,
* housingMedianAge,
* totalRooms,
* totalBedrooms,
* population,
* households,
* medianIncome,
* medianHouseValue,
* oceanProximity.

## Loss function

To quantify the extent to which the predicted response value for a given observation is close to the true response value for that observation, the most commonly used measure in regression setting is the mean squared error (MSE).

## Quality of fit

The loss function it is used for [parameter estimation](https://en.wikipedia.org/wiki/Parameter_estimation) to map an [event](https://en.wikipedia.org/wiki/Event_(probability_theory)) or values of one or more variables onto a [real number](https://en.wikipedia.org/wiki/Real_number) intuitively representing some “cost” associated with the event. [[13]](#footnote-13)

With loss function we denote the measure of how different the prediction of a hypothesis is from the true outcome.

We use a nonnegative loss function to measure the discrepancy between the predicted label and the true label . In the regression task we define the quadratic loss that is the squared distance between and

When then otherwise If and *c* is large then also tend to be large.

y . {\displaystyle y.} The [risk](https://en.wikipedia.org/wiki/Risk_(statistics)) associated with hypothesis h ( x ) {\displaystyle h(x)} is then defined as the [expectation](https://en.wikipedia.org/wiki/Expected_value) of the loss function:

*The risk function , or the expected loss, becomes the mean squared error of the estimate with respect to the probability distribution*

in regression we refer to the estimation of a probability density as the loss function as the Euclidian distance in a function space , indeed:

And the risk function becomes the squared distances between our target variable and predicted values

The L2 Loss [Mean Square Error (MSE)](https://medium.freecodecamp.org/machine-learning-mean-squared-error-regression-line-c7dde9a26b93) is used as loss function to perform this Rigde regression.

## Empirical Risk Minimization

The empirical risk minimization defines a family of learning algorithms used to output some predictor minimizing the training error given a non-negative real-valued loss function :

The empirical risk minimization principle states that the learning algorithm should choose a hypothesis which minimizes the empirical risk[[14]](#footnote-14).

## Test Error and training error

**(**generalization error**)**

est MSE, for a given value x0, can always be decomposed into the sum of three fundamental quantities: the variance of f(x0), the squared bias of f(x0) and the variance of the error terms e. Where, e is the irreducible error, about which we discusses earlier. So, lets see more about bias and variance.

Estimate the predictive power of a predictor, we use the test error which allows to perform the validation of the algorithm. It is given by

,

The training error is given to fit the model on training set and perform the model selection and it is

The bias is an error term that stems from a mismatch between the model class and the under-lying data distribution, and is typically monotonically non-increasing as a function of the complexity of the model.

Where high complexity models could have very low bias, but high variance.

Whereas low complexity models have high bias, but low variance.

Model variance increases and bias decreases monotonically with model complexity

BIAS VARIANCE TRADE OFF

In any case the prediction will not be perfectly accurate because of the kind of errors that can occur. Total error is given by bias, variance and the irreducible error. Since the latter cannot be improved the variance and bias have to be optimized.

To find

A general rule is that, **as a statistical method tries to match data points more closely or when a more flexible method is used, the bias reduces, but variance increases.**

There is always a trade-oﬀ between these values because it is easy to obtain a method with extremely low bias but high variance (for instance, by drawing a curve that passes through every single training observation) or a method with very low variance but high bias (by ﬁt a horizontal line to the data). The challenge lies in ﬁnding a method for which both the variance and the squared bias are low.

The bias is the difference between the true population parameter and the expected estimator:

It measures the accuracy of the estimates.



Graph: the optimal choose for trade-off[[15]](#footnote-15)

In Regularization regression we use the hyper parameter α to add a bias , prevent the model from real, it will have better regression on the testing set, it means it will not overfit in the training set.

Hight bias if the function is non linear or can be nearly linear then my error will be small

If we change regression, the fit can change significally, it is instable and it leads to variance error. High bias and high variance.

## Regressions – Linear, Ridge, Lasso

Ridge and Lasso regression are part of the family of ERM algorithms. They modify the standard linear regression by introducing a positive constant as a regularization parameter. Indeed the objective function to minimize under these solutions is

Rss + penalty

and the penalty is different for the two methods.

Starting from the classical linear model we have

Let be the data domain and

. The linear predictor is a linear function of the data points

, where and where . The Regression with square loss, we want the linear predictor to be as good as possible, for istance, we can look at the risk that would be the base optimal

The empirical risk minimization to is

Since we can rewrite these terms in vector notation, we have

For the vector of predictions and the vector of real labels and for .

We can rewrite everything in matrix notation, let be the design matrix with features and observations we get from data points, that are rows of , and therefore the vector becomes

and applying the ERM we derive

The solution to the ERM is the minimization of this convex function applying the Euclidian distance.

To solve the problem in linear regression we can use closed form solution, or the gradient descend.

If is a non-singular matrix[[16]](#footnote-17), and the conditions of the general position holds, the solution of the ERM is

In some cases, a linear regression performs good in finding a fit on the training data, but it gives a non-accurate estimate on different data. The reason why it occurs it is because of multicollinearity of the prediction vectors (as known as non- orthogonality)[[17]](#footnote-18)

More in general with large or small, the risk that the model can overfit[[18]](#footnote-19) the data is high. The OLS estimator therefore is unbised but have a huge variance and it is not stable.

*To overcome this problem, Ridge and Lasso regression help to prevent over-fitting which results from simple linear regression.*

We introduce a regularized parameter which adds some bias[[19]](#footnote-20) whereas pushing the variance down.

This occurs to find the best trade-off between bias and variance to get to that sweet spot for having good predictive performance.

The two methods work similarly but lead to different results, this happens because of the divergent formulas.

### Ridge solution

Ridge regression uses the penalty multiplied by the square of the magnitude of the coefficients, also known as L2 regularization.

The ERM functional of Ridge regression is

,

For , so the solution leads the linear regression, for the coefficient tend to a zero vector and the line becomes flatter, shrinking the linear regression solution towards to zero.

To optimize the objective function, we take the gradient as before and solve for to find a suitable value

The new estimated parameter becomes

This is the so called closed-form solution and is the one measuring the stability of the procedure.

### Lasso solution

Least Absolute Shrinkage and Selection Operator, or simply Lasso, is slightly different from the previous because the penalty is multiplied by the [absolute value](https://www.statisticshowto.com/integer/#abs) of the magnitude of coefficients, also known as L1 regularization

,

For

The Lasso procedure encourages simple, sparse models[[20]](#footnote-22), indeed some coefficients can become zero and be eliminated from the model, this way performing a feature selection.

The shrinkage amount is given by the value of [tuning parameter](https://www.statisticshowto.com/tuning-parameter/) . If increase, we have some parameters go straightway to zero.

Λ (lambda) provides a trade-off between balancing RSS and magnitude of coefficients.

The optimization of a non-differentiable function as Lasso solution is done by a proximal gradient descend approach.

The first step is to take the gradient descend for current vector and form a new vector :

Where is the step size and is the moment we are considering.

Then solve the proximal regularize problem for as follows:

This is a scaler minimization problem indeed we can rewrite it as :

Since we have an absolute value for we consider two cases:

**Case 1**

Differentiate with respect to and solve:

Therefore, since we have the non-negativity constraint over

If ,

Otherwise

**Case 2**

Differentiate with respect to and solve for it:

If

Otherwise

The three solutions are also known as the “soft threshold” operation,

There is some shrinkage going on in this three cases, w is the shank version of z

where the value taken by w is the shrinkage of z

The update rule of the algorithm is therefore

DA COMPLETARE

The common point of these two methods is that adding the regularization parameter to the cost function the algorithm is forced to pick the lowest weights, indeed the goal is to ensure a small coefficient through this regularization parameter.

The main difference is that many coefficients are exactly zeroed under lasso, which is never the case in ridge regression where there is not any elimination of coefficients.

### Cross-validation

Cross-validation (CV) is one of the techniques used to test the effectiveness of a machine learning models, it is also a re-sampling procedure used to evaluate a model if we have a limited data[[21]](#footnote-23).

In case of limited data, this approach can present a high bias, since there is a substantial loss of data in the training set, but if data is huge and our test sample and train sample has the same distribution then this approach is acceptable.

### K-fold

K-fold cross-validation is a non-parametric method for evaluating the accuracy of a predictive rule.[[22]](#footnote-24)

It ensures that every observation from the original dataset has the chance of appearing in training and test set

for model selection

Hyperparameters whose value must be determined before the training phase can start

The computed error on the testing part of each fold is

is the dataset divided in subsets , where is the testing part.

We want to estimate the for a fixed hyperparameter.

And the CV final error is the egress of the errors

given the choice of two predictors, it repeatedly picks the most accurate of the two. Fix a training\test set .

So for tuning the hyperparameter

we look on

where

Estimate risk on all S

ALGORITMO POI LO AGGIUSTO IO VISIVAMENTE

This is the algorithm[[23]](#footnote-25)

for p in 1:P:

for k in 1:K:

keep fold k as hold-out data

use the remaining folds and λ = λp to estimate β^ridge

predict hold-out data: ytest,k=Xtest,kβ^ridge

compute a sum of squared residuals: SSRk = ||y − ytest, k||2

end for k

average SSR over the folds: SSRp=1K∑k=1KSSRk

end for p

choose optimal value: λopt = argminpSSRp

Moreover the choose of k

he k value must be chosen carefully for your data sample.

A poorly chosen value for k may result in a mis-representative idea of the skill of the model, such as a score with a high variance (that may change a lot based on the data used to fit the model), or a high bias, (such as an overestimate of the skill of the model).

Always remember, a lower value of k is more biased, and hence undesirable. On the other hand, a higher value of K is less biased, but can suffer from large variability. It is important to know that a smaller value of k always takes us towards validation set approach, whereas a higher value of k leads to LOOCV approach.

HOW TO READ BIAS VARIANCE TRADE OFF

After k-fold cross validation, we’ll get *k* different model estimation errors (e1, e2 …..ek). In an ideal scenario, these error values should sum up to zero. To return the model’s bias, we take the average of all the errors. Lower the average value, better the model.

Similarly for calculating the model variance, we take standard deviation of all the errors. A low value of standard deviation suggests our model does not vary a lot with different subsets of training data.

We should focus on achieving a balance between bias and variance. This can be done by reducing the variance and controlling bias to an extent. It’ll result in a better predictive model. This trade-off usually leads to building less complex predictive models as well.

### Principal component analysis

Principal Component Analysis is a mathematical technique, unsupervised used for dimensionality reduction. Its goal is to reduce the number of features through a combinations of the original data variables. In this way keeping most of the original information.

Let’s identify the principal components and interpret their relationship to the original variables. The linear coefficients for the PCs (sometimes called the “loadings”) are shown in the columns of the Eigenvectors table.

Unsupervised algorithms, principal component analysis (PCA)

The features are selected on the basis of variance that they cause in the output.

The feature that causes highest variance is the first principal component.

The feature that is responsible for second highest variance is considered the second principal component, and so on.

It is important to mention that principal components do not have any correlation with each other.

It can be seen that first principal component is responsible for 72.22% variance

A general rule of thumb is to take number of principal of principal components that contribute to significant variance and ignore those with diminishing variance returns. A good way is to plot the variance against principal components and ignore the principal components with diminishing values as shown in the following graph:[[24]](#footnote-26)

VIDEO

We increase the number of features in a intelligent way, because pca keep the most informative.

This way bias is increased but we can have a more stable prediction, it increases the bias but reduces the variance.

One of the points is a linear (ridge)predictor that we have learned.

But basically, to inmprove the stability we can use ridge regression as well.

Therefore, ridge regression is a soft PCA regression in fact. They both intend to solve the multi-collinearity in order to improve the model egress.

In the PCA analysis negative values of loadings of variable in the components of the PCA means the existence of an inverse correlation between the factor PCA and the variables.

# Proof of a technical result

The dataset presents features that cannot be compared in a linear Euclidian space; therefore, geometry is not working properly on this row data. Indeed, in order to learn the algorithm, we need to encode the features and raise them to a homogeneous level, so we can compare them (ex. Longitude and households are real numbers however they represent different meanings)

Before performing the analysis and regression, pre-procession of data is necessary.

The dataset contains 20640 observations and 10 features for each house including the median house value which is the target value that we are trying to predict. Firstly, we create the two constants of labels and data domain:

= median\_house\_value

= data frame

The missing values are handled to avoid errors in the execution of the code.There is a categorical feature which represents the distance from the ocean, we assign a numerical value to each category of labels. The dataset is randomized and outliers are dropped.

Features such as garden, household size, neighborhood satisfaction and schools were considered statistically insignificant in a hedonic model as demonstrated by Berna and Craig [12].

## Normalization

We used min-max normalization for rescaling the da-tasets into a normalized form

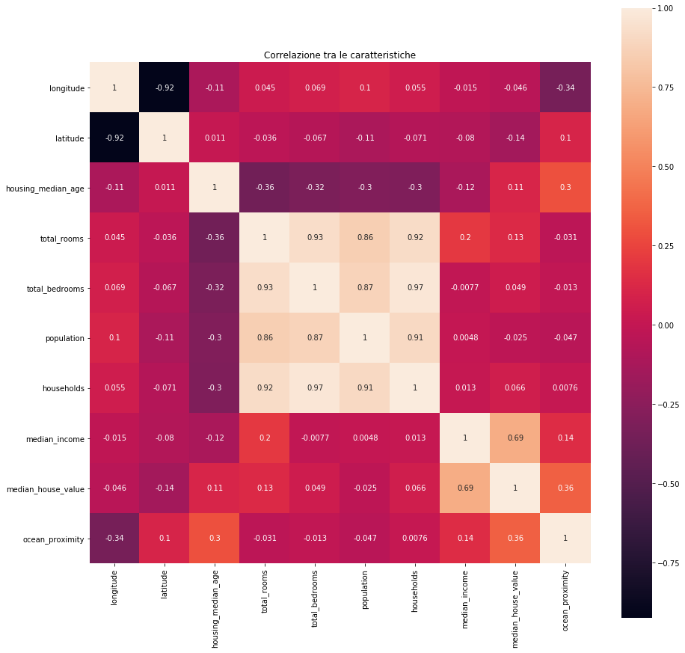
Standardized features for implementing algorithms thar use eucleadian distance

To implement pca

It is important to standardize the features by removing the mean and scaling to unit variance.

The L1 (Lasso) and L2 (Ridge) regularizers of linear models assume that all features are centered around 0 and have variance in the same order.

Correlation



*Correlation matrix,*

provided enough evidence that there exists statistical re-lationship between the variables. Therefore, our dataset was suitable for decomposition into its principal components. A principal component analysis (PCA) was necessary to find the core components of the datasets, increase convergence speed and eliminate collinearity.

The PCA algorithm trans-formed the data into some smaller and more meaningful com-ponents which were the true representation of the attributes. In other words, new variables were derived from the datasets. PC1 is the first principal component; it is the linear combina-tion of variables with most possible variance in the datasets. The second one, PC2 covers the next variance. A new princi-pal component is assumed to be uncorrelated with all previ-ous components. Mathematically; ܥܲ݊௜=(ܽ௜ଵܸ ଵ)+(ܽ௜ଶܸ ଶ)+⋯+(ܽ௜௡ܸ ௡). (2) Where nPis the number of components, ܽ௜ଵ.....ܽ௜௡are the component weights and V1.....Vn the variables.

## Model Tuning

### Scoring

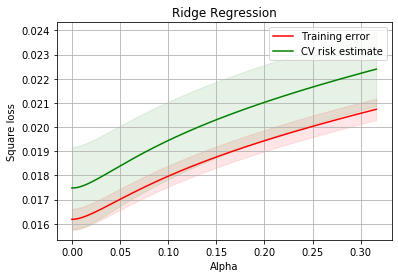
We use the RMS as scoring, so the ---

CORRELATION

Finally, looking at Pearson’s correlation matrix a features reduction is performed by taking away the most correlated feature with a egressio of |0.75|

### Choosing the set of parameters alpha

In order to obtain a reasonable amount of information to determine a certain where is the function that minimize the training error, we use a logarithmic range, .

Training size m is bigger than the in order to avoid underfitting.

As we can see the optimal value for the hyperparameter is around 0.01, after that the squared lost increases.

## Optimization algorithm

### Cross- validate risk estimate

Lasso

Cross-validated risk estimate: 0.022512047975181457

Ridge

Cross-validated risk estimate: 0.02580007952062598

[the plot]

as we see the generalized variance is smaller then in training set as this is the k-fold egressio that allows to have this results.

Large a ---------------small a

Hight bias low bias

Low varance high variance

Ex w hat = 0 rss

We perform an analysis of different tunning parameters alpha and the relative mean squared error, comparing the Lasso and Ridge (Cholensky)



The best egre value is 0.2 more or less for ridge regression, then the error increases (>0.022) for values alpha > 0.2

In lasso regression this increase is smoother, and mse is more stable for alpha in between 0.1 and 1.

This is what we could expect if there are feature highly correlated and that are not crucial for the regression.

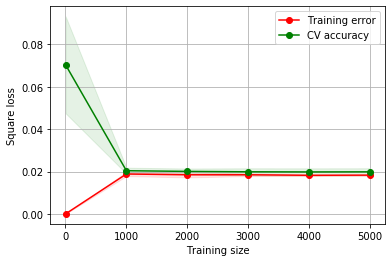
On the other hand R2 is speculat to the mse and tells u show well the model fits the data



The coefficient magnitude is shown in this graph



Median income has a huge power in the prediction, it means that it drives the values.

****

### PCA

Cells that are highly correlated cluster together

Differences between the 1st pc zxex are more important then the differences between the 2nd pc

Interpretation of the graph:

<https://online.stat.psu.edu/stat505/lesson/11/11.4>

# Some critical considerations

## Outliers problem

Even if ridge egression s modelled in a way to deal with multicollinearity and outliers, it still being sensitive to outliers. In our work there were some outliers that were pushing the fit in a wrong direction.

<https://iopscience.iop.org/article/10.1088/1742-6596/890/1/012150>

<https://iopscience.iop.org/article/10.1088/1742-6596/890/1/012150/pdf>

As the model complexity increases, the models tends to fit even smaller deviations in the training data set. Though this leads to overfitting, lets keep this issue aside for some time and come to our main objective, i.e. the impact on the magnitude of coefficients. This can be analysed by looking at the data frame created above.

It is clearly evident that the **size of coefficients increases exponentially with increase in model complexity**. I hope this gives some intuition into why putting a constraint on the magnitude of coefficients can be a good idea to reduce model complexity.

What does a large coefficient signify? It means that we’re putting a lot of emphasis on that feature, i.e. the particular feature is a good predictor for the outcome. When it becomes too large, the algorithm starts modelling intricate relations to estimate the output and ends up overfitting to the particular training data.

Lasso can set some coefficients to zero, thus performing variable selection, while ridge regression cannot.

This way Lasso performs better in terms of reducing the variance in models with a lot of features (ritondanti)

In contrast Ridge regression performs better in models where a lot of features are important

# Bibliographical references

* Calhoun C. A., 2003, “Property Valuation Models and House Price Indexes for The Provinces of Thailand: 1992 –2000”, Housing Finance International, 17(3): 31 –41.

Frew J. and B. Wilson, 2000, “Estimation The Connection Between Location and Property Value”, Essay in Honor of James A.Graaskamp, Boston, MA: Kluwer Academic Publishers

* Oladunni, Timothy & Sharma, Sharad. (2016). Hedonic Housing Theory – A Machine Learning Investigation. 10.1109/ICMLA.2016.0092.
* Limsombunchai, Visit & Gan, Christopher & Lee, Minsoo. (2004). House Price Prediction: Hedonic Price Model vs. Artificial Neural Network. American Journal of Applied Sciences. 1. 10.3844/ajassp.2004.193.201.
* Manjula, R & Jain, Shubham & Srivastava, Sharad & Kher, Pranav. (2017). Real estate value prediction using multivariate regression models. IOP Conference Series: Materials Science and Engineering. 263. 042098. 10.1088/1757-899X/263/4/042098.
* Dubin, Robin. (1998). Predicting House Prices Using Multiple Listings Data. The Journal of Real Estate Finance and Economics. 17. 35-59. 10.1023/A:1007751112669.
* Xin, Seng & Khalid, Kamil. (2018). Modelling House Price Using Ridge Regression and Lasso Regression. International Journal of Engineering & Technology. 7. 498. 10.14419/ijet.v7i4.30.22378.
* Dobriban, Edgar & Wager, Stefan. (2015). High-Dimensional Asymptotics of Prediction: Ridge Regression and Classification. The Annals of Statistics. 46. 10.1214/17-AOS1549.
* Zvi Griliches, 1991. "[Hedonic Price Indexes and the Measurement of Capital and Productivity: Some Historical Reflections](https://ideas.repec.org/h/nbr/nberch/5976.html)," [NBER Chapters](https://ideas.repec.org/s/nbr/nberch.html), in: [Fifty Years of Economic Measurement: The Jubilee of the Conference on Research in Income and Wealth](https://ideas.repec.org/b/nbr/nberbk/bern91-1.html), pages 185-206, National Bureau of Economic Research, Inc.

Santarelli M. 2020

<https://www.noradarealestate.com/blog/housing-market-predictions/>

- Frew J. and B. Wilson, 2000, “Estimation The Connection Between Location and Property Value”, Essay in Honor of James A.Graaskamp, Boston, MA: Kluwer Academic Publishers

Ray,2018 <https://www.analyticsvidhya.com/blog/2018/05/improve-model-performance-cross-validation-in-python-r/>

Liu, Sifan & Dobriban, Edgar. (2020). Ridge Regression: Structure, Cross-Validation, and Sketching.

Rosen S., 1974, “Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition”, Journal of Political Economics, 82: 34 – 55

# Copyright

We declare that this material, which We now submit for assessment, is entirely our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of our work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by us or any other person for assessment on this or any other course of study.

1. Santarelli, 2020 [↑](#footnote-ref-1)
2. Source: International Investor Survey [↑](#footnote-ref-2)
3. Source: United States Census Bureau <https://www.census.gov/quickfacts/fact/table/US/HSG445218> [↑](#footnote-ref-3)
4. Oladunni, Timothy & Sharma, Sharad. (2016) [↑](#footnote-ref-4)
5. Limsombunchai et al. , 2004 [↑](#footnote-ref-5)
6. the so-called Sales comparison approach [↑](#footnote-ref-6)
7. Introduced by Hoerl and Kennard (1970), [↑](#footnote-ref-7)
8. Riccardo A. Maronna [↑](#footnote-ref-8)
9. Griliches 1991 [↑](#footnote-ref-9)
10. Oladunni, Timothy & Sharma, Sharad. (2016) [↑](#footnote-ref-10)
11. In other words the algorithm has not enough data to train and be approximated [↑](#footnote-ref-11)
12. <https://openreview.net/pdf?id=HklRwaEKwB> [↑](#footnote-ref-12)
13. <https://en.wikipedia.org/wiki/Loss_function> [↑](#footnote-ref-13)
14. V. Vapnik (1992). [<http://papers.nips.cc/paper/506-principles-of-risk-minimization-for-learning-theory.pdf> *Principles of Risk Minimization* for Learning Theory.*]* [↑](#footnote-ref-14)
15. Source : researchgate.net [↑](#footnote-ref-15)
16. This happens if the data points span [↑](#footnote-ref-17)
17. <https://www.math.arizona.edu/~hzhang/math574m/Read/RidgeRegressionBiasedEstimationForNonorthogonalProblems.pdf> [↑](#footnote-ref-18)
18. Overfitting: the algorithm performs very good on training data but cannot be generalized to a new bunch of data. [↑](#footnote-ref-19)
19. Bias is how well the fit correspond to the true value [↑](#footnote-ref-20)
20. <https://www.statisticshowto.com/lasso-regression/> [↑](#footnote-ref-22)
21. <https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f> [↑](#footnote-ref-23)
22. <https://arxiv.org/pdf/1909.11696.pdf> [↑](#footnote-ref-24)
23. <https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net> [↑](#footnote-ref-25)
24. <https://stackabuse.com/implementing-pca-in-python-with-scikit-learn/> [↑](#footnote-ref-26)