Bereket Eshete Kebede
UID - U00827234

M.S. Student
University of Memphis, Fall 2022
COMP 7745 Machine Learning

Implementing a Linear Regression Model
Implementing a Logistic Regression Model

Instructor: Prof. Zahangir Alom
Assignment #1 Report

Table of contents

COMP 7745 Machine Learning Assignment

Sept 19, 2022

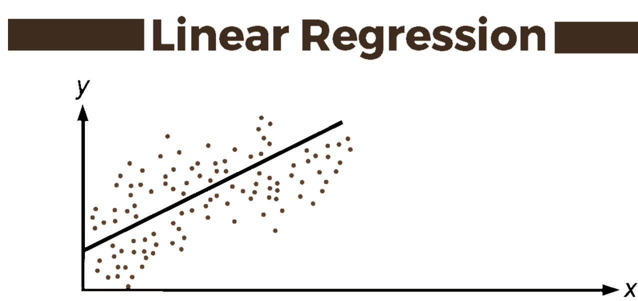
Implementing a Linear Regression Model

Introduction

In this report we will Implement a linear regression model to predict the house price for the provided datasets. Linear regression analysis is used to predict the value of a variable based on the value of another variable. If you ask a prospective homeowner to describe their ideal home, they usually won't start by talking about how high the basement ceiling is or how close the house is to an east-west railroad. However, the dataset for this report shows that factors other than the number of beds and a white picket fence have a significant impact on price negotiations. We will estimate the final price of each residential property in Ames, Iowa, given the 79 explanatory factors that describe (nearly) every feature of residential properties there [1].

Methodology

Linear regression is a linear approach for modeling the relationship between a scalar response and one or more explanatory variables [2]. Linear Regression is the simplest algorithm in machine learning and it can be trained in different ways [3]. We will first split up our data into an array that contains the features to train on, and a y array with the target variable, in our case the Price column. We will toss out columns if it only has text info that the linear regression model can't use.



Many regression models rely on distance metrics to determine the convergence to the best result. Usually the metrics used are the Mean Average Error (MAE), the Mean Squared Error (MSE) or the Root Mean Squared Error (RMSE) [3,4].

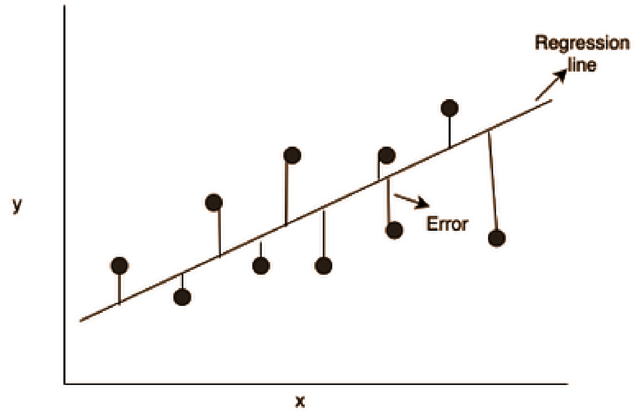
$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i^{real} - y_i^{pred}|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i^{real} - y_i^{pred})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i^{real} - y_i^{pred})^2}$$

Model descriptions

The regression model operates by minimizing the error between the point distribution and the linear prediction line as shown in the figure below [3]



Experiment and Results

(a) Database

train.csv - the training set
test.csv - the test set
data_description.txt - full description of each column, originally prepared by Dean De Cock but lightly edited to match the column names used here.

SalePrice - the property's sale price in dollars. This is the target variable that you're trying to predict.	**BsmtUnfSF**: Unfinished square feet of basement area
MSSubClass: The building class	**TotalBsmtSF**: Total square feet of basement area
MSZoning: The general zoning classification	**Heating**: Type of heating
LotFrontage: Linear feet of street connected to property	**HeatingQC**: Heating quality and condition
LotArea: Lot size in square feet	**CentralAir**: Central air conditioning
Street: Type of road access	**Electrical**: Electrical system
Alley: Type of alley access	**1stFlrSF**: First Floor square feet
LotShape: General shape of property	**2ndFlrSF**: Second floor square feet
	LowQualFinSF: Low quality finished square feet (all

LandContour: Flatness of the property	floors)
Utilities: Type of utilities available	**GrLivArea**: Above grade (ground) living area square feet
LotConfig: Lot configuration	**BsmtFullBath**: Basement full bathrooms
LandSlope: Slope of property	**BsmtHalfBath**: Basement half bathrooms
Neighborhood: Physical locations within Ames city limits	**FullBath**: Full bathrooms above grade
Condition1: Proximity to main road or railroad	**HalfBath**: Half baths above grade
Condition2: Proximity to main road or railroad (if a second is present)	**Bedroom**: Number of bedrooms above basement level
BldgType: Type of dwelling	**Kitchen**: Number of kitchens
HouseStyle: Style of dwelling	**KitchenQual**: Kitchen quality
OverallQual: Overall material and finish quality	**TotRmsAbvGrd**: Total rooms above grade (does not include bathrooms)
OverallCond: Overall condition rating	**Functional**: Home functionality rating
YearBuilt: Original construction date	**Fireplaces**: Number of fireplaces
YearRemodAdd: Remodel date	**FireplaceQu**: Fireplace quality
RoofStyle: Type of roof	**GarageType**: Garage location
RoofMatl: Roof material	**GarageYrBlt**: Year garage was built
Exterior1st: Exterior covering on house	**GarageFinish**: Interior finish of the garage
Exterior2nd: Exterior covering on house (if more than one material)	**GarageCars**: Size of garage in car capacity
MasVnrType: Masonry veneer type	**GarageArea**: Size of garage in square feet
MasVnrArea: Masonry veneer area in square feet	**GarageQual**: Garage quality
ExterQual: Exterior material quality	**GarageCond**: Garage condition
ExterCond: Present condition of the material on the exterior	**PavedDrive**: Paved driveway
Foundation: Type of foundation	**WoodDeckSF**: Wood deck area in square feet
BsmtQual: Height of the basement	**OpenPorchSF**: Open porch area in square feet
BsmtCond: General condition of the basement	**EnclosedPorch**: Enclosed porch area in square feet
BsmtExposure: Walkout or garden level basement walls	**3SsnPorch**: Three season porch area in square feet
BsmtFinType1: Quality of basement finished area	**ScreenPorch**: Screen porch area in square feet
BsmtFinSF1: Type 1 finished square feet	**PoolArea**: Pool area in square feet
BsmtFinType2: Quality of second finished area (if present)	**PoolQC**: Pool quality
BsmtFinSF2: Type 2 finished square feet	**Fence**: Fence quality
	MiscFeature: Miscellaneous feature not covered in other categories
	MiscVal: $Value of miscellaneous feature
	MoSold: Month Sold
	YrSold: Year Sold
	SaleType: Type of sale
	SaleCondition: Condition of sale

Table 1. Dataset features and the descriptions

Data preprocessing:

We need to fill in the gaps where there is no data available in the columns. Hence, we create dummy variables for the categorical features and replace the numeric missing values (NaN's) with the mean of their respective columns.

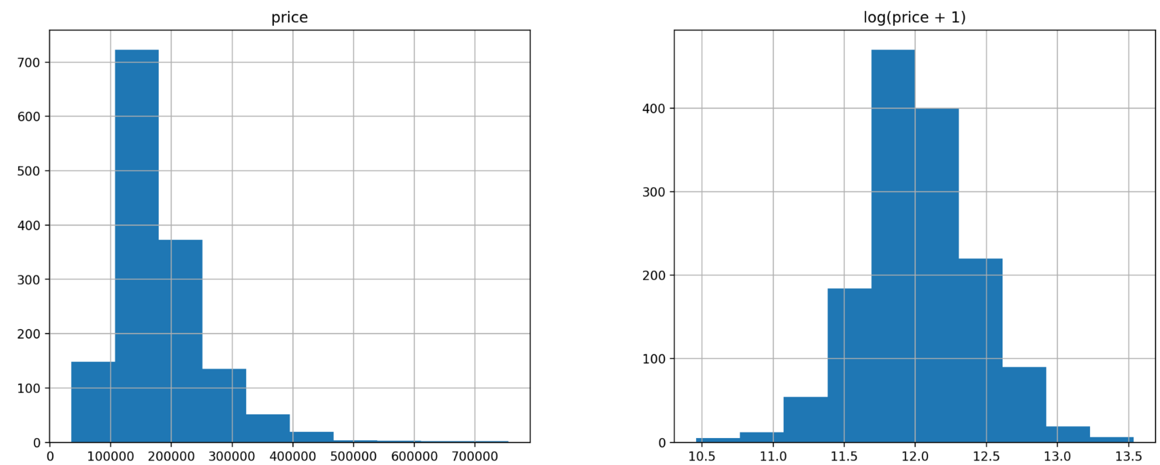


Figure 1. Data histogram plot

(b) Training and testing logs

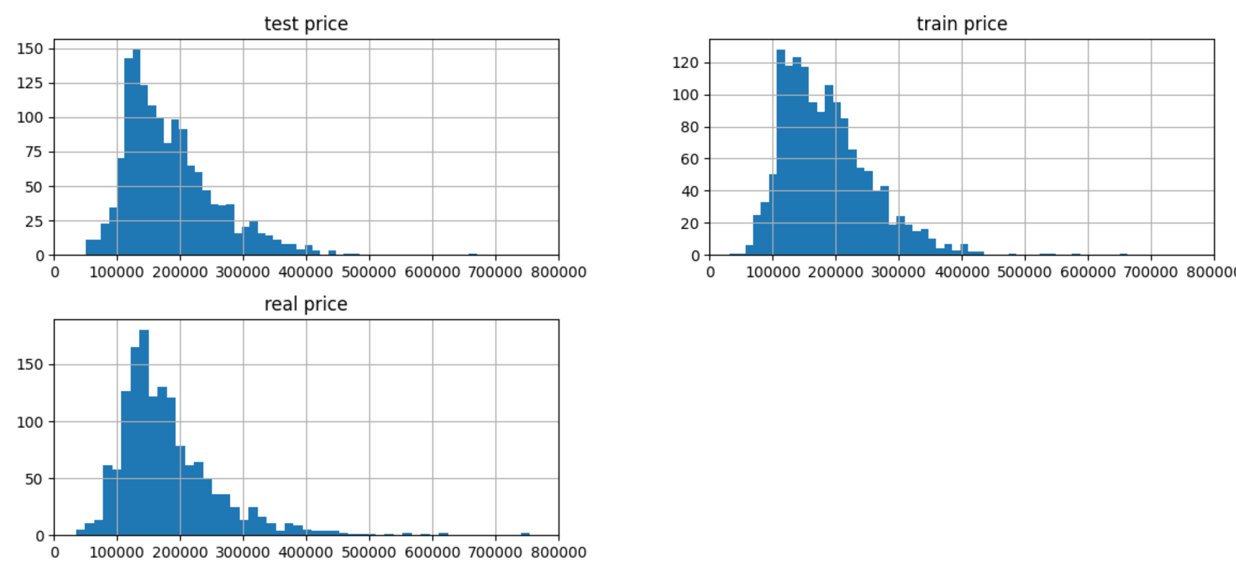


Figure 2. Training and testing plot

(c) Discussion and comparison

Table 2. Analysis of the results with Means Squared Error (MSE), absolute errors, and Root MSE (RMSE).

MAE:	**79489.39**
MSE:	**11379036010.16**
RMSE:	**106672.56**

Conclusion

Linear regression can be used to predict the price of houses with a good accuracy as shown in figure 2.

References

[1] https://www.kaggle.com/c/house-prices-advanced-regression-techniques
[2] https://en.wikipedia.org/wiki/Linear_regression
[3] https://community.cloudera.com/t5/Community-Articles/Understanding-Linear-Regression/ta-p/281391
[4] https://towardsdatascience.com/comparing-robustness-of-mae-mse-and-rmse-6d69da870828

Implementing a Logistic Regression Model

Introduction

Linear Regression is used to handle regression problems whereas Logistic regression is used to handle the classification problems. Linear regression provides a continuous output but Logistic regression provides a discrete output.

- Elastic Net - In statistics and, in particular, in the fitting of linear or logistic regression models, the elastic net is a regularized regression method that linearly combines the L_1 and L_2 penalties of the lasso and ridge methods.
- Lasso Net - is a method for feature selection in neural networks, to enhance interpretability of the final network.
- Ridge Net - Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where the independent variables are highly correlated.

Methodology & Model descriptions

In elastic Net Regularization we added the both terms of L1 and L2 to get the final loss function. This leads us to reduce the following loss function. Where alpha is between 0 and 1. when alpha = 1, It reduces the penalty term to L1 penalty and if \alpha = 0, it reduces that term to L2 penalty.

$$L_{elastic-Net}\left(\hat{\beta}\right) = \left(\sum\left(y - x_i^J\hat{\beta}\right)^2\right)/2n + \lambda\left((1-\alpha)/2 * \sum_{j=1}^{m}\hat{\beta}_j^2 + \alpha * \sum_{j=1}^{m}\left\|\hat{\beta}_j\right\|\right)$$

Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares,. α>=0 is a complexity parameter that controls the amount of shrinkage: the larger the value of A. α, the greater the amount of shrinkage and thus the coefficients become more robust to collinearity. Ridge regression is an L2 penalized model. Add the squared sum of the weights to the least-squares cost function.

$$\min_{w}||Xw - y||_2^2 + \alpha||w||_2^2$$

Lasso regression, is a linear model that estimates sparse coefficients. Mathematically, it consists of a linear model trained with $\ell 1$ prior as regularizer. The objective function to minimize is:

$$\min_{w} \frac{1}{2n_{samples}} ||Xw - y||_2^2 + \alpha ||w||_1$$

Experiment and Results

(a) Database

Similar to Task (A), train.csv and test.csv

(b) Training and testing logs

Table 3. Metric values for the three regression models

	RMSE	MSE	MAE
Ridge	0.131	0.018	0.085
Lasso	0.123	0.015	0.080
Elastic	0.204	0.042	0.149

(c) Discussion and comparison

In this report we covered the common linear regression models (Ridge, Lasso and ElasticNet). We saw the representation used by the model. We discussed rules of thumb to consider when preparing data for use with linear regression and finally we saw how to evaluate a linear regression model.

Conclusion

We implemented three logistic regression models and evaluated their performance metrics.

References

[1] https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/
[2] Elastic Net, https://en.wikipedia.org/wiki/Elastic_net_regularization
[3] Lasso Net, https://lassonet.ml/

[3] Ridge Net,
https://www.kaggle.com/code/faressayah/practical-introduction-to-10-regression-algorithm/notebook#%E2%9C%94%EF%B8%8F-Ridge-Regression