Single Output Regression Analysis
Data Science Analysis Project

Prasanth Tata
EP20BTECH11023

Gaureesh K
EP20BTECH11005

May 2, 2023

Contents

Abstract

Machine learning has emerged as a promising field that provides accurate predictions and insights based on complex data. With an abundance of machine learning models available, it is important to compare their performance to identify the most appropriate model for a particular task. We consider a dataset of housing having 17 features like area, existence of a garage, pool, attic etc. The price of the house is predicted using different machine learning algorithms such as linear regression, lasso regression, ridge regression, elastic net regression, support vector machine, random forest. To compare the efficiency and accuracy of each alogrithms different evaluation metrics like R^2 score, RMS error are calculated and compared across the different regression models.

1 INTRODUCTION

Machine learning is a field where we provide machines with the ability to learn automatically from past data. They are very helpful in automating task by finding patterns and regularities in data. We generally consider that data follows an arbitrary function $g(x)$ which we normally do not know. To predict future data we require knowledge of $g(x)$. We assume a prediction function $f(x)$ and by looking at the existing data we modify the parameters of $f(x)$ to make it as close to $g(x)$ as possible.

Regression is a supervised learning algorithm used to predict a continuous output variable based on input features. It involves training a model on a dataset with known input-output pairs and using it to predict the output for new, unseen data. Regression models learn the relationship between the input variables and the output variable and use this relationship to make predictions. The goal of regression in machine learning is to minimize the difference between the predicted output and the actual output for each input. There are various types of regression algorithms, including linear regression, logistic regression, and decision tree regression, each with its own advantages and disadvantages depending on the problem at hand. Regression is a fundamental tool in machine learning and is used in a wide range of applications, such as financial forecasting, medical diagnosis, and image analysis.

When developing a regression model in machine learning, it is crucial to compare different models to choose the one that best suits the problem at hand. One common approach to comparing models is to use a metric such as Root mean squared error (RMSE), which measures the average squared difference between the predicted and actual values. Lower RMSE values indicate better performance. Such metrics help assess how well the model generalizes to new data.

2 Code

The github repository for the code developed for this project can be accessed from the URL : https://github.com/prashtata/DSA-Project

3 Dataset used

The dataset from ParisHousing.csv, has 10000 rows and 17 columns. The column 'price' is the label we want to predict. More details of the feature names and definition is given in the assignment question. Calculating the correlation of each feature with price, we get

	Correlation with price
garage	-0.017229
isNewBuilt	-0.010643
made	-0.007210
hasYard	-0.006119
hasPool	-0.005070
basement	-0.003967
hasStorageRoom	-0.003485
cityCode	-0.001539
hasGuestRoom	-0.000644
attic	-0.000600
floors	0.001654
hasStormProtector	0.007496
cityPartRange	0.008813
numberOfRooms	0.009591
numPrevOwners	0.016619
squareMeters	0.999999
price	1.000000

Figure 1: Correlation of different features with price

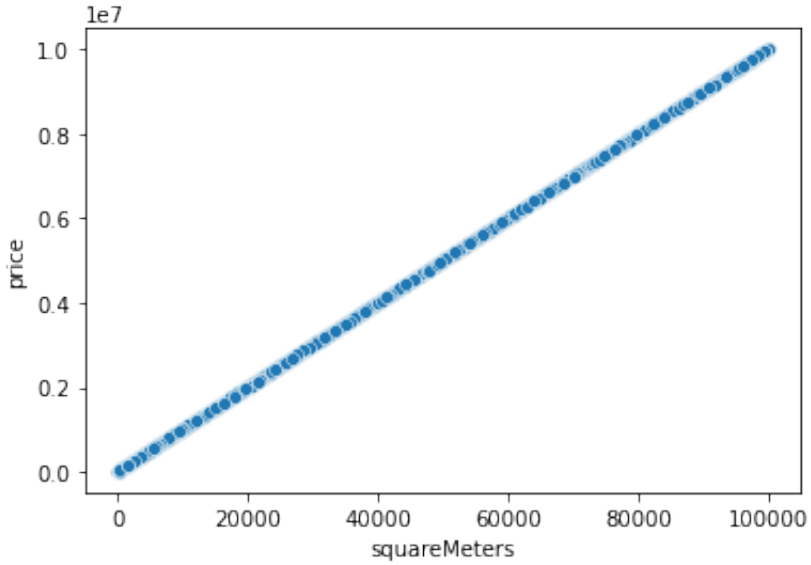

Figure 2: Price vs squareMeters

From this we can see that other than the squareMeters feature (which varies linearly with price), all other features barely have any correlation with price. We will later compare the accuracy of keeping all features and if we drop all columns other than squareMeters.

4 Diffferent regression algorithms used

We use different regression algorithms to predict the prices and compare the end result using different parameters.

4.1 Linear Regression

linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables).If there is 1 output(dependent) variable it is called simple linear regression, whereas if there are more than 1 dependent variable, it is called multiple linear regression. Linear regression algorithm is used if the labels are continuous and there is a linear relation between dependent and independent variables.
For a simple linear regression, the representation is $y = b + c$
where b and c are the parameters to be determined. The cost function is defined : (Predicted output – actual output)^2.

4.2 Lasso Regression

Another popular linear ML regression that only requires one input variable is called Lasso (Least Absolute Shrinkage and Selection Operator). To prevent prediction errors, lasso regression penalises the sum of coefficient values. 'Shrinkage' is a technique used in lasso regression to lower the determination coefficients towards zero. In order to make the regression coefficients precisely fit with different datasets, lasso regression (L1 regulatization) is used to lower the coefficients. The lasso approach is also used in Data Mining for regression in addition to ML. This approach is widely used in cases where high multicollinearity in the dataset, which means that the independent variables are highly correlated to each other. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters) by regularization (penalising number of regression coefficients)

4.3 Ridge Regression

Ridge regression is another model tuning method that is used to analyse any data that suffers from multicollinearity where two or more independent variables in a data frame have a high correlation with one another in a

regression model. This technique carries out L2 regularisation. Predicted values differ much from real values when the problem of multicollinearity arises, least-squares are unbiased, and variances are significant.
The cost function for Ridge regression is $||Y - X(\theta)||^2 + \lambda||\theta||^2$
where the second term is the regularization to reduce number of parameters,

4.4 Elastic Net Regression

Elastic net linear regression regularises regression models by using the penalties from the lasso and ridge procedures. In order to improve the regularisation of statistical models, the strategy combines the lasso and ridge regression approaches.

4.5 SVM

The primary goal of SVM is to find the best hyperplane that separates the different classes in the given data.

In simple terms, SVM finds a decision boundary (hyperplane) that maximally separates the data points of different classes by finding the closest data points to the boundary (support vectors). The distance between the support vectors and the decision boundary is called the margin, and SVM aims to maximize this margin.

SVM works by mapping the input data into a high-dimensional feature space, where it becomes easier to find a hyperplane that separates the data. The choice of the kernel function used for mapping the data into the feature space is critical to the performance of SVM. Some popular kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid.

SVM can be used for both binary and multi-class classification problems, and it is also useful for regression problems.

4.6 Random Forest

Another popular approach for non-linear regression in machine learning is random forest. A random forest uses many decision trees to forecast the outcome as opposed to decision tree regression (single tree). With the help of this approach, a decision tree is constructed using k randomly chosen data points from the given dataset. Several decision trees are then modeled that predict the value of any new data point. A random forest method will

anticipate many output values because there are numerous decision trees. To get the final result for a new data point, you must calculate the average of all the anticipated values.
The sole disadvantage of utilising a random forest approach is that more training data is needed. This occurs because this technique maps a large number of decision trees, which uses greater computational resources.Hence bootstrapping can be used here to reuse the data.

5 Model Comparision

We use the following metrics to compare the performance of different regression models.

- Time taken to train
- Mean absolute error
- Mean absolute error as a percent of mean value of price
- Root mean squared error
- Root mean squared error as a percent of standard deviation of price
- R^2 score

5.1 R^2 Score

R-squared is a goodness-of-fit measure for linear regression models. This statistic shows the proportion of the dependent variable's variance that the independent variables account for collectively. R-squared provides a straightforward 0–100 percent scale to quantify the strength of the association between your model and the dependent variable.

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$

Usually, the larger the R2, the better the regression model fits your observations. However, this guideline has some exceptions. Residues in an unbiased model are dispersed at random in the vicinity of zero. Hence non-random residual patterns can have a bad fit even when the R^2 is high. This is why it is important to check how the residue varies while using R^2 score as a metric to judge the goodness of fit of a model.

6 Results

Training the different regression models on the dataset, we are able to obtain the following results:

6.1 Linear Regression

Using default values,

```
Linear Regression

Time taken:  0.04593825340270996
Mean absolute error:  1517.5594934334927
Mean absolute error percent:  0.030391017140118465
Root mean squared error:  1935.099961280118
Root mean squared error percent:  0.067251120701741
R2 score:  0.9999995419971762
```

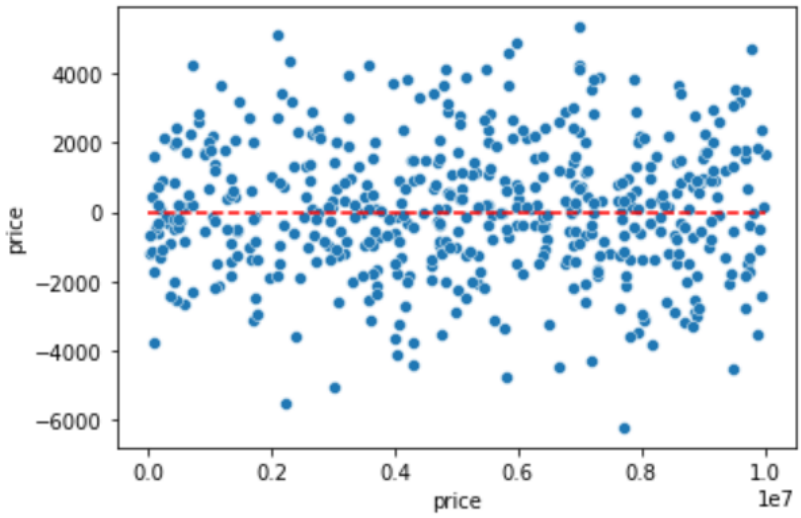

Figure 3: Linear Regression

6.2 Lasso Regression

Using k fold cross validation with epsilon = 0.0000001, number of alphas = 1000 and k=10,

```
Lasso Regression

Time taken:  1.9324064254760742
Mean absolute error:  1517.581499733077
Mean absolute error percent:  0.03039145784364986
Root mean squared error:  1934.5985001878867
Root mean squared error percent:  0.06723369327105748
R2 score:  0.9999995422345189
```

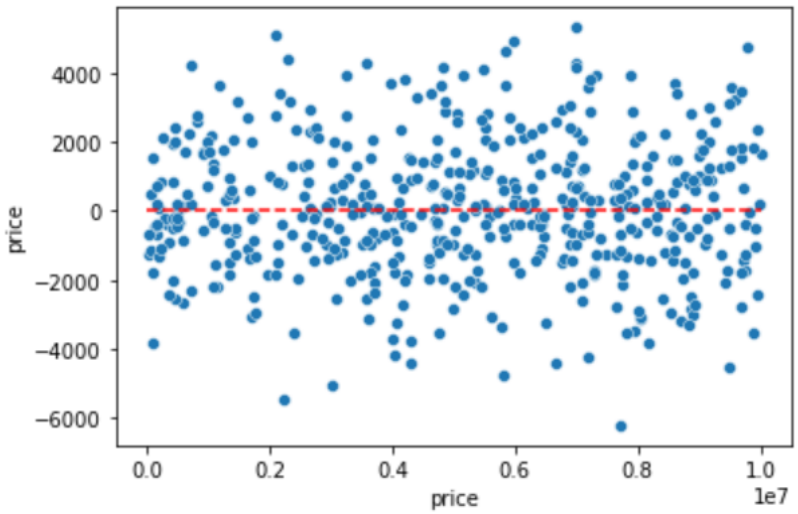

Figure 4: Lasso Regression

6.3 Ridge Regression

Using k fold cross validation over alphas = [0.01,0.1,0.5,1,5,10] with k=10,

```
Ridge Regression

Time taken:  0.2652907371520996
Mean absolute error:  1517.6801955317346
Mean absolute error percent:  0.030393434349824174
Root mean squared error:  1935.1217081119196
Root mean squared error percent:  0.06725187647603671
R2 score:  0.999999541986882
```

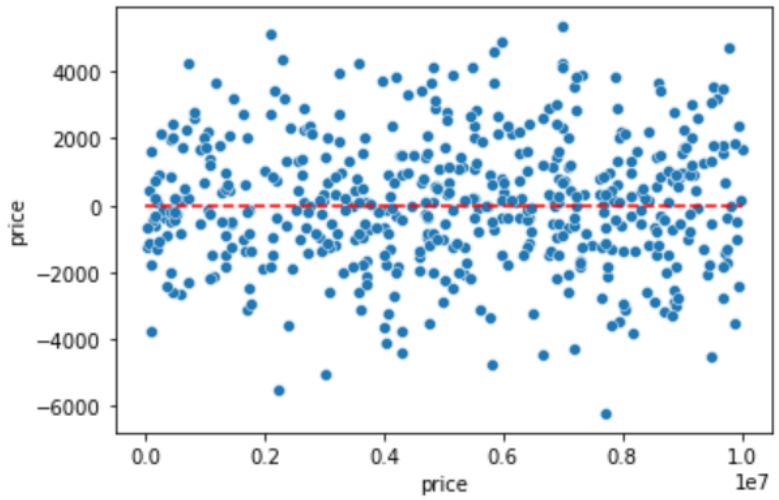

Figure 5: Ridge Regression

6.4 Elastic Net

Using k fold cross validation with l1 ratio = [.1, .5, .7, .9, .95, .99, 1], epsilon = 0.0000001, number of alphas=1000 and k=10,

```
Elastic Net

Time taken:  13.97571063041687
Mean absolute error:  1517.581499733077
Mean absolute error percent:  0.03039145784364986
Root mean squared error:  1934.5985001878867
Root mean squared error percent:  0.06723369327105748
R2 score:  0.9999995422345189
```

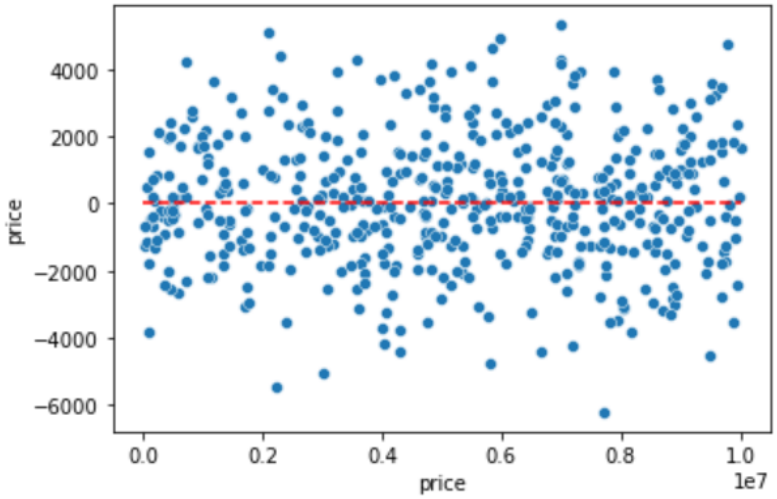

Figure 6: Elastic Net

6.5 SVM (Support Vector Machine)

6.5.1 With default values

Using default values

```
SVM(default)

Time taken:  5.388606548309326
Mean absolute error:  2449238.46311274
Mean absolute error percent:  49.04904778677714
Root mean squared error:  2863245.568402843
Root mean squared error percent:  99.50724881003232
R2 score:  -0.0027173775166557945
```

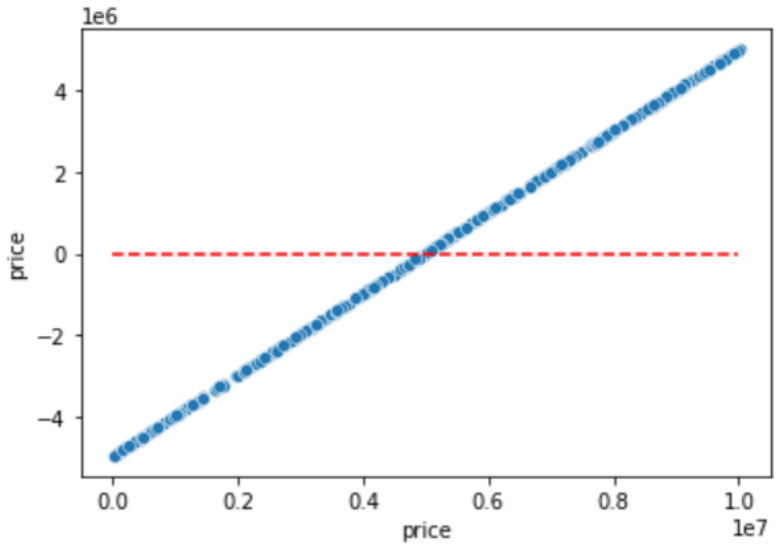

Figure 7: SVM using default values

6.5.2 With grid search

```
SVM(with CV)

Time taken:  674.5323822498322
Mean absolute error:  1610.0496456782728
Mean absolute error percent:  0.032243247523392374
Root mean squared error:  2044.2739714977135
Root mean squared error percent:  0.0710452784638959
R2 score:  0.9999994876518482
```

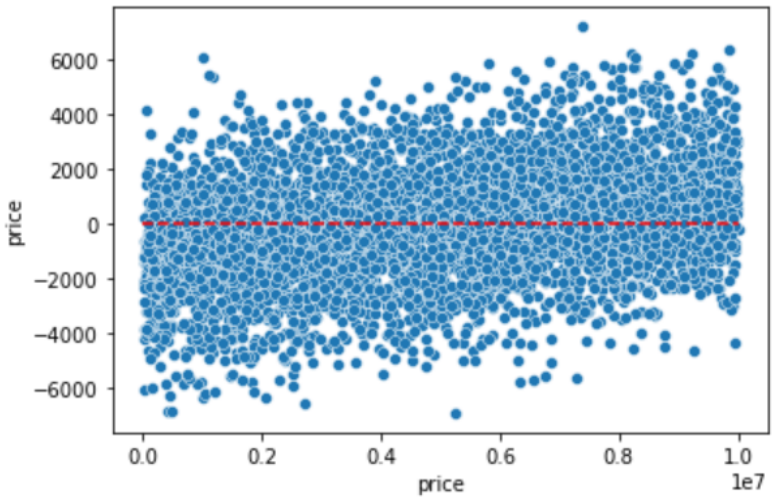

Figure 8: SVM with grid search

6.6 Random Forest Regression

Using default values,

```
Random Forest

Time taken:  5.983237266540527
Mean absolute error:  1137.4712873667697
Mean absolute error percent:  0.022779277873675736
Root mean squared error:  1429.5651183401205
Root mean squared error percent:  0.049682113714110736
R2 score:  0.999999749449738
```

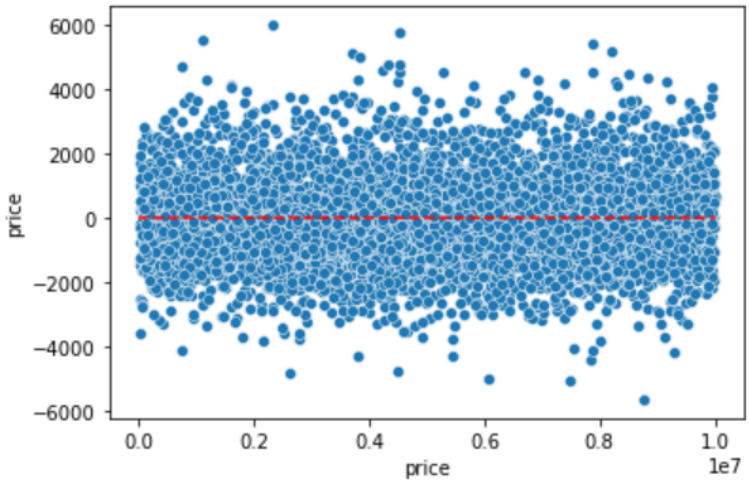

Figure 9: Random Forest Regression

7 Conclusion

Comparing all the data (sorted by time taken to train),

	Time	MAE	MAE %	RMSE	RMSE %	R2 Score
Linear	0.006979	1.488131e+03	0.029802	1.909196e+03	0.066351	1.000000
Ridge	0.253320	1.488130e+03	0.029802	1.909228e+03	0.066352	1.000000
Lasso	1.969372	1.488098e+03	0.029801	1.909444e+03	0.066360	1.000000
SVM Default	5.188857	2.461044e+06	49.285462	2.856104e+06	99.259054	-0.000081
Random Forest	5.983237	1.137471e+03	0.022779	1.429565e+03	0.049682	1.000000
Elastic Net	15.687243	1.488098e+03	0.029801	1.909444e+03	0.066360	1.000000
SVM CV	674.532382	1.610050e+03	0.032243	2.044274e+03	0.071045	0.999999

Figure 10: Comparison of evaluation metrics for different Regression Models

From this, we can see that in most algorithms, we get similar error values (SVM default is probably has a lot of error because SVM's accuracy is highly dependent on the C and gamma values), but the time for linear regression is the least by a large margin. Therefore, among the different models tested here on the given dataset, linear regression is objectively the best choice due to its extremely high training speed for similar quality.