Ridge regression is a type of linear regression that penalizes ridge coefficients. This technique can be used to reduce the effects of multicollinearity in ridge regression, which may result from high correlations among predictors or between predictors and independent variables.

What is Ridge Regression?

Ridge regression is a powerful technique in machine learning that addresses the issue of overfitting in linear models.
In linear regression, we aim to model the relationship between a response variable and one or more predictor variables. However, when there are multiple variables that are highly correlated, the model can become too complex and prone to overfitting. This is where ridge regression comes into play.

Example to illustrate this. Imagine you are working on a housing dataset where you want to predict the price of a house based on various features such as size, number of bedrooms, location, and age. In this case, it's highly likely that some of these features are correlated with each other. For instance, the size of the house and the number of bedrooms might be strongly correlated. This multicollinearity can lead to unstable and unreliable coefficient estimates in a traditional linear regression model.

To combat this, ridge regression introduces a penalty term to the error function, known as the ridge penalty or L2 regularization term. This penalty shrinks the coefficients towards zero, reducing their magnitudes. By adding this penalty, ridge regression prevents the coefficients from becoming too large, effectively reducing the complexity of the model. This helps to mitigate the problem of overfitting by striking a balance between capturing the underlying patterns in the data and avoiding excessive reliance on individual data points.

Returning to our housing example, ridge regression would help in handling multicollinearity. By applying the ridge penalty, the coefficients for the highly correlated features (e.g., size and number of bedrooms) would be penalized, ensuring that the model does not assign too much weight to any one feature. This regularization effect enhances the stability and generalization capability of the model.

Moreover, ridge regression offers some resistance to outliers, which are extreme data points that deviate significantly from the majority of the data.

Outliers can have a disproportionate impact on traditional linear regression models, pulling the estimated coefficients towards them and affecting the model's overall performance. In contrast, ridge regression reduces the sensitivity to outliers by shrinking the coefficients, making the model more robust to these influential data points.

However, it's important to note that while ridge regression can provide some resilience to outliers, it is not specifically designed to handle them.

For datasets with a substantial presence of outliers, other methods such as robust regression techniques or the L1 regularization (used in LASSO regression) might be more appropriate.

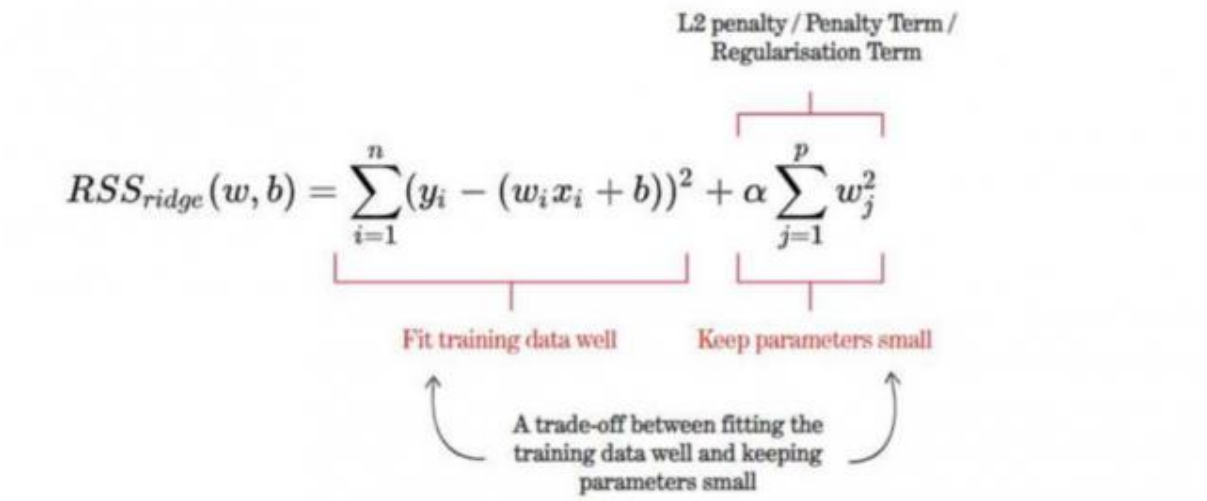
How does Ridge Regression work?

Ridge regression works by adding a penalty term to the cost function, the penalty term being proportional to the sum of the squares of the coefficients.

The penalty term is called the **L2 norm**. The result is that the optimization problem becomes easier to solve and the coefficients become smaller.

This penalty term encourages the model to find a balance between fitting the training data well and having low complexity. As a result, ridge regression can help to improve the generalizability of a machine learning model.
The cost function for ridge regression looks like this. You may note that the cost function comprises two functions. The first one is the cost function same as the one used for the linear regression model. This term ensures that the training data fits well. The second term is called the L2 penalty or regularization term. The goal of this term is to keep the parameters small.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$



where y is the predicted value, x is the input value, β is the coefficient, and λ is the penalty term. As you can see, the penalty term is added to the error term. In ridge regression, we want to minimize both the error and the size of the coefficients. By adding the penalty term, we are encouraged to find a balance between these two objectives.

Advantages & disadvantages of Ridge Regression

Ridge regression penalizes the sum of the squared coefficients, or beta values, in order to reduce the variance of the estimates. It shrinks the coefficients and thus reduces the standard errors. The penalty term serves to reduce the magnitude of the coefficients, and it also helps to prevent overfitting. As a result, Ridge regression can provide improved predictive accuracy. This ultimately results in more stable and accurate predictions.

Ridge regression also has the ability to handle nonlinear relationships between predictor and outcome variables better than linear regression.

Ridge regression advantages.

1. it is more robust to collinearity than least-squares/linear regression.
2. ridge regression does not require the data to be perfectly normalized.
3. ridge regression can be applied even when the number of variables is greater than the number of observations.

However, ridge regression also has some disadvantages.

1. it can be computationally expensive if the data set is large.
2. it can be difficult to interpret the results of ridge regression because the Ridge term or L2 norm modifies the coefficients. This is because the cost function contains a quadratic term, which makes it more difficult to optimize.
3. ridge regression does not provide an exact solution and instead only provides a closed-form approximation. This can make it difficult to interpret the results of the model.
4. ridge regression can offer some degree of resistance to outliers due to the regularization effect, but it is not as robust as other dedicated outlier-robust regression techniques.

Lasso Regression

Lasso regression, also known as **L1 regularization**, is a linear regression method that uses regularization to prevent overfitting and improve model performance.

It works by adding a penalty term to the cost function that encourages the model to select only the most important features and set the coefficients of less important features to zero. This makes Lasso regression a popular method for feature selection and high-dimensional data analysis.

What's Lasso Regression?

Lasso regression is a machine learning algorithm that can be used to perform linear regression while also reducing the number of features used in the model. Lasso stands for **least absolute shrinkage and selection operator**. Pay attention to the words, "least absolute shrinkage" and "selection". We will refer to it shortly. Lasso regression is used in machine learning to prevent overfitting. It is also used to select features by setting coefficients to zero. Lasso regression is also called **L1-norm regularization.** In L1 regularization, a penalty term is added to the cost function that is proportional to the sum of the absolute values of the coefficients. This encourages the model to select only the most important features and set the coefficients of less important features to zero. Compared to other regularization methods, such as **Ridge regression** that uses L2 regularization, Lasso regression has the advantage of producing sparse solutions, where only a subset of the features are used in the model. This makes Lasso regression a popular method for feature selection and high-dimensional data analysis.
One limitation of Lasso regression is that it tends to work better when the number of features is smaller than the number of samples. This is because Lasso regression can completely eliminate some features from the model by setting their coefficients to zero, which can be problematic when the number of features is large.

Lasso regression is an extension of linear regression in the manner that a regularization parameter multiplied by the summation of the absolute value of weights gets added to the loss function (ordinary least squares) of linear regression. Lasso regression is also called **regularized linear regression**. The idea is to induce the penalty against complexity by adding the regularization term such that with increasing value of the regularization parameter, the weights get reduced (and, hence penalty induced) to keep the overall goal of the minimized sum of squares. The hypothesis or the mathematical model (equation) for Lasso regression is the same as linear regression and can be expressed as the following. However, what is different is loss function.

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

$$\textit{Parameters: } \theta = \{\theta_0, \theta_1, \theta_2, \ldots . \theta_n\}$$

$$\textit{Features: } x = \{x_0, x_1, x_2, \ldots . x_n\}$$

Fig 1. Lasso Regression Hypothesis Function

Here is the loss function of LASSO regression. Compare it with the loss function of linear regression.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}) + \frac{\lambda}{m} \sum_{j=1}^{n} |\theta_j|$$

Fig 2. Lasso Regression Loss Function

Compare it with the linear regression loss function.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

Fig 3. Linear Regression Loss Function

You may note that in Lasso regression's loss function, there is an extra element such as the following:

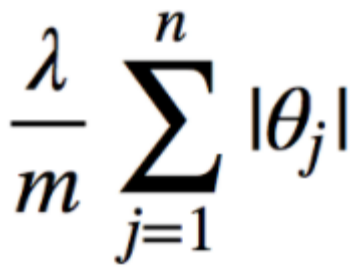
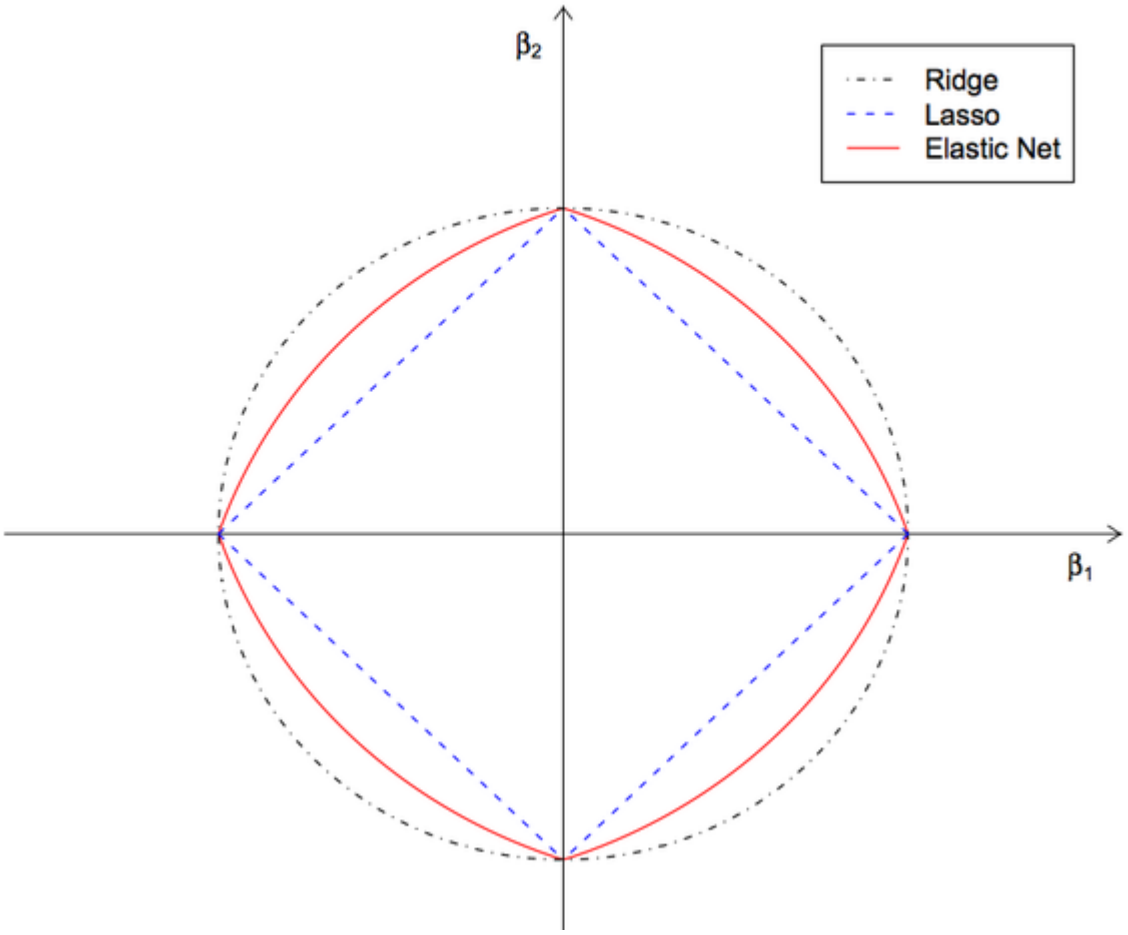


Fig 4. Regularization parameter with absolute summation of weights values

The equation in fig 4 represents the regularization parameter λ and summation of absolute values of weights. “m” represents the constant. The increasing value of the regularization parameter means increasing regularization strength, the absolute values of weights would need to decrease (shrink) to keep the overall value of the loss function minimized. The optimization of the Lasso loss function results in some of the weights becoming zero and hence can be seen as a method of selection of the features. Pay attention to the usage of words, **shrinkage, selection**, and **absolute.** This is why LASSO is termed as **Least absolute shrinkage and selection operator**. Optimizing the LASSO loss function does result in some of the weights becoming zero. Thus, some of the features will be removed as a result. This is why LASSO regression is considered to be useful as a **supervised feature selection** technique.

Elasticnet Regression

Evaluate the model by calculating the mean square error. Let’s get started step by step.



Resource: https://www.oreilly.com/library/view/machine-learning-with/9781787121515/5c5ec380-d139-49a5-99b1-3ce32ae5bd6f.xhtml

What is the ElasticNet Regression?

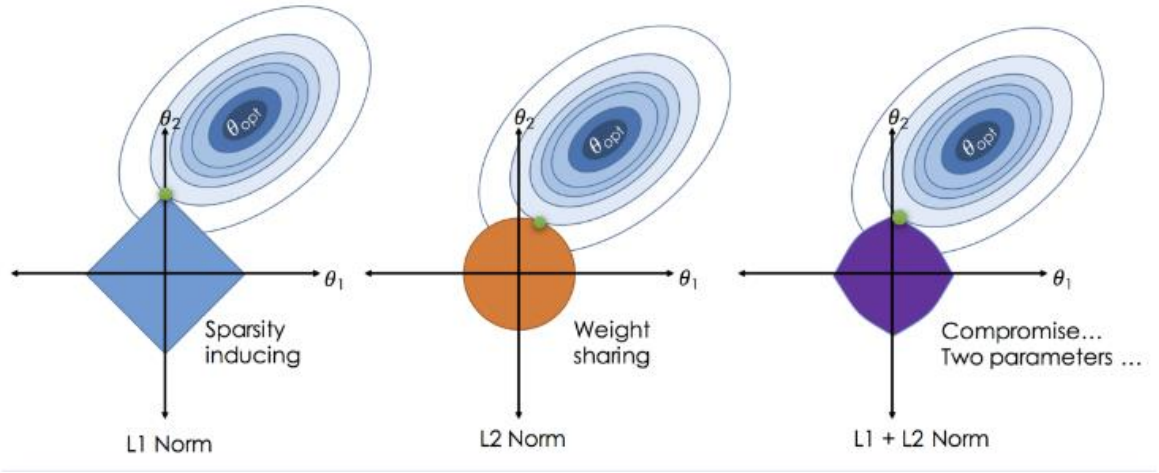
The main purpose of ElasticNet Regression is to find the coefficients that minimize the sum of error squares by applying a penalty to these coefficients. ElasticNet combines L1 and L2 (Lasso and Ridge) approaches. As a result, it performs a more efficient smoothing process. In another source, it is defined as follows:

Elastic Net first emerged as a result of critique on Lasso, whose variable selection can be too dependent on data and thus unstable. The solution is to combine the penalties of Ridge regression and Lasso to get the best of both worlds.

Features of ElasticNet Regression

- It combines the L1 and L2 approaches.
- It performs a more efficient regularization process.
- It has two parameters to be set, λ and α.

The elastic net method improves on lasso's limitations, i.e., where lasso takes a few samples for high dimensional data, the elastic net procedure provides the inclusion of "n" number of variables until saturation. In a case where the variables are highly correlated groups, lasso tends to choose one variable from such groups and ignore the rest entirely.



Differences between L1, L2, and L1+L2 Norm

ElasticNet Regression Model

Elastic Net aims at minimizing the following loss function:

$$L_{enet}(\hat{\beta}) = \frac{\sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2}{2n} + \lambda\left(\frac{1-\alpha}{2}\sum_{j-1}^{m}\hat{\beta}_j^2 + \alpha\sum_{j-1}^{m}|\hat{\beta}_j|\right),$$

ElasticNet Mathematical Model

The terms used in the mathematical model are the same as in Ridge and Lasso Regression.

Modeling with Python

Now let's build a `ElasticNet Regression` model on a sample data set. And then let's calculate the square root of the model's `Mean Squared Error` This will give us the model error.

First of all, we import the libraries necessary for modeling as usual.

Then we do data reading and some data editing operations.

With ElasticNet regression, we set up the model on the train set.

I do not go into concept details such as what is fit, what is a train set.

According to the variables in the data set we have, we find the variable coefficients in the ElasticNet model as follows.

We found the constant of the ElasticNet regression model to be **-6,46** with the following function.

Prediction

Now let's make the model prediction under normal conditions without specifying any parameters. We can see the first 10observations of the model prediction for the train set as follows.

Likewise, we can see the first 10observations of the model prediction for the test set as follows.

Then we saved the values we predicted over the test set in a cluster named y_pred. And we found the RMSE value as **357,16** as a result of the following calculation.

As a result, we found the R-Squared score as 0,41. The R-squared score is the percentage of the change in the dependent variable explained by the independent variables.

In other words, we can say that independent variables in ElasticNet Regression Model explain 41.07% of the change in dependent variables for this data set.

What is R-Squared?

R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable. So, if the R2 of a model is 0.50, then approximately half of the observed variation can be explained by the model's inputs.

Model Tuning

ElasticNetCV method to find the optimum lambda value.

Accordingly, we find the alpha value as **5230,76**.

Afterward, we can find the constant of the model established with ElasticNetCV as follows.

We can find the coefficients of the variables of the model established with ElasticNetCV as follows.

Then we rebuild the Adjusted ElasticNet model with this optimum alpha value. Then we print the predicted values from the test set into y_pred. As a result, we find the RMSE value as **394,15.**

Finally

First, we examined what is ElasticNet Regression in this blog post. Then we talked about the features and basics of ElasticNet Regression. Mathematically, we examined the model of this algorithm. Then we set up the model under the current conditions and calculate the error value. In the Model Tuning part, we calculated the corrected error value by calculating the optimum alpha value with ElasticNetCV and rebuilding the corrected model according to this alpha value.