**Linear Regression, Ridge Regression, Lasso Regression, and Polynomial Regression:**

**Linear Regression:**

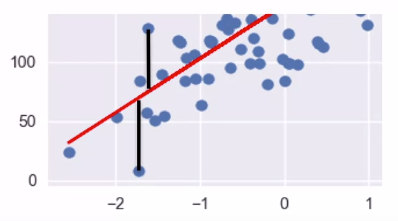
A Linear model is a model that uses a sum of weighted variables to predict a target value of a particular instance. E.g. predicting the price of a house.

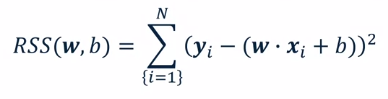
E.g.:

So, the model takes on a linear form of where the instance feature vector is

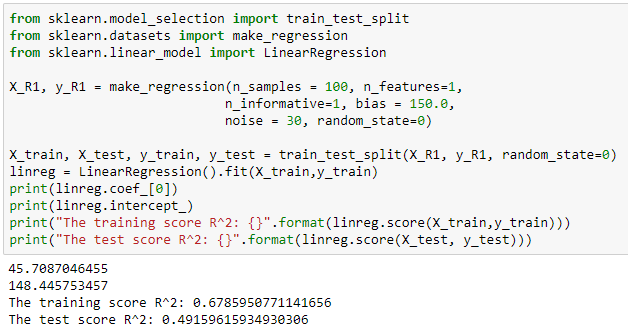
The gradient of the model is given by

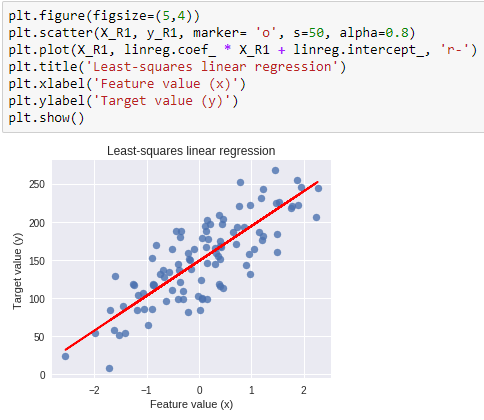
**How does the algorithm find the values of w and b?**

We use the **Least-Squares** method to find the optimum values of w and b to ensure that the model generalizes to the training data well. This find the values of the parameter estimators that return the smallest value of mean squared error of all the features, essentially the mean vertical distance between the training points and the line given by both w and b

The most popular way of finding w and b is to use the **Least-squares** method: where yi is the training known target value, and xi is its features.

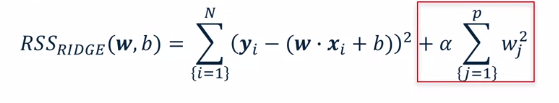
We try and find the values of w and b that give the **minimum** value of this **RSS** equation. We then use these values of w and b to create a prediction model in the form y = mx + c.





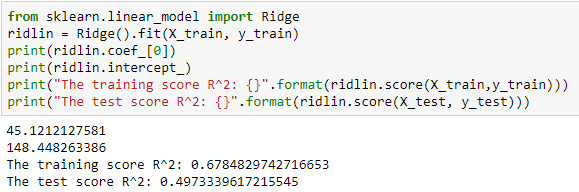
**Ridge Regression:**

This method is very similar to the least-squares method but adds a penalty for **large variations in w parameters.** This type of penalty is known as **regularization α,** and prevents overfitting by restricting the model, this typically reduces the complexity.

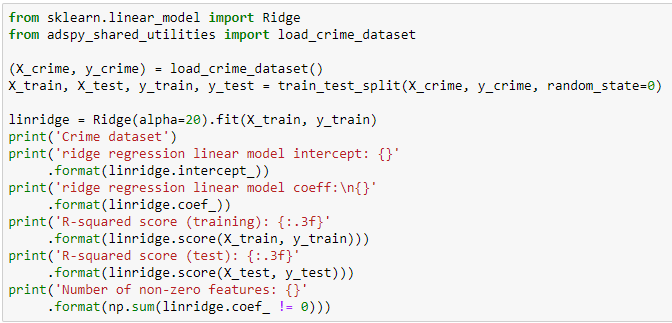


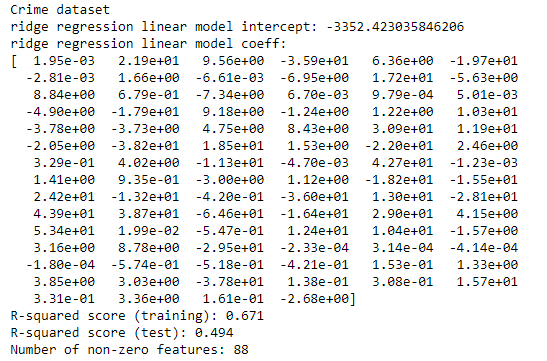
Once the parameter w and b are learned the model works in the same way. The regularization parameter, in RSS, ensures that the model returns a value for w that is smaller than a normal linear regression would. The larger the value of alpha the smaller the gradient of the model will be and the simpler the model.

**The model performs better than a normal linear regression when there are many features**



Below I have used the crime data set.



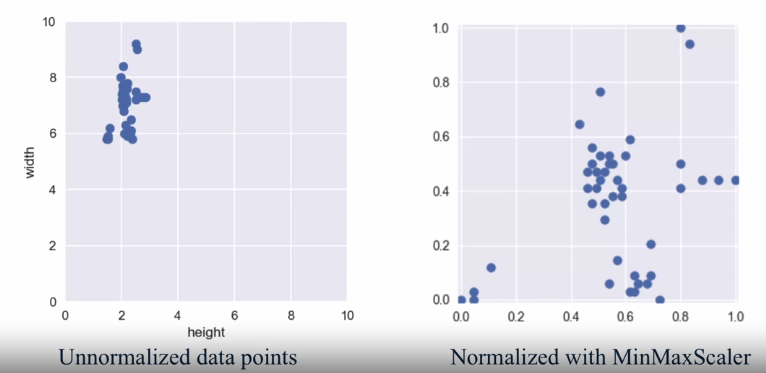


**Feature Pre-processing and Normalization:**

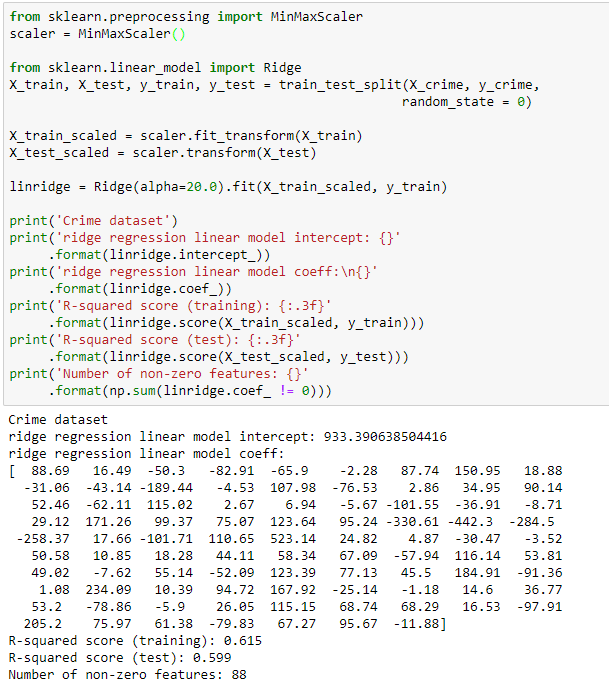
Scaling the data should be done to improve performance on any model that uses some form of **Euclidean distance**. E.g. if there are two categories in your data, one being in a range of values from 0-1 and the other being in a range -10000 to 10000, then the larger feature will have much more of an impact over the values w and b. If we scale the data, both values will be between an equal range and therefore have equal impact on the output of RSS.

**MinMax:** this scales the data to have values only between 0 and 1.



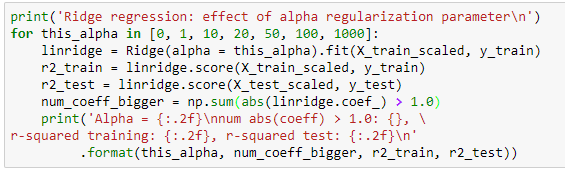


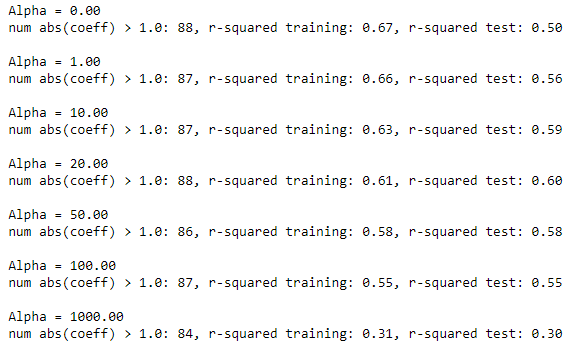
We only fit the training data to the scaler. If we fit the scaler to the test data, we’re assuming we know information about it, the test data is supposed to be new data instances the model will encounter when deployed. If testing data get into the scaler this is known as **data leakage.**



This shows increase scores from the previous code that didn’t use a **scaler**.

Going back to using ridge regression, we can see what changing the parameter α does to the model:

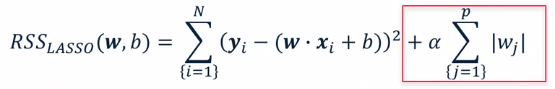




We can see that for this data set the model is at an optimal point when the alpha value is around 20.

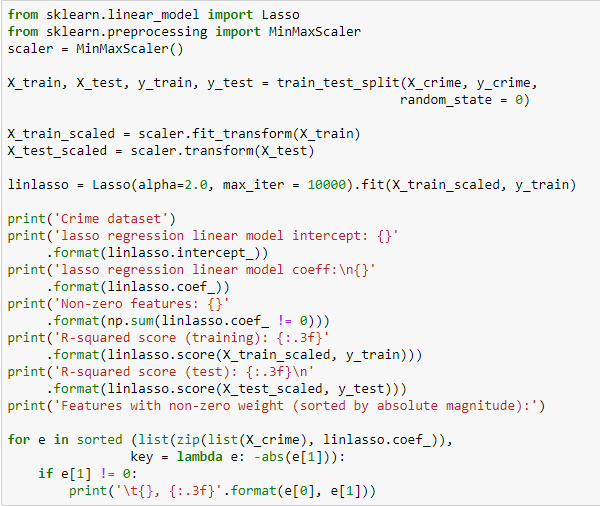
**Lasso Regression:**

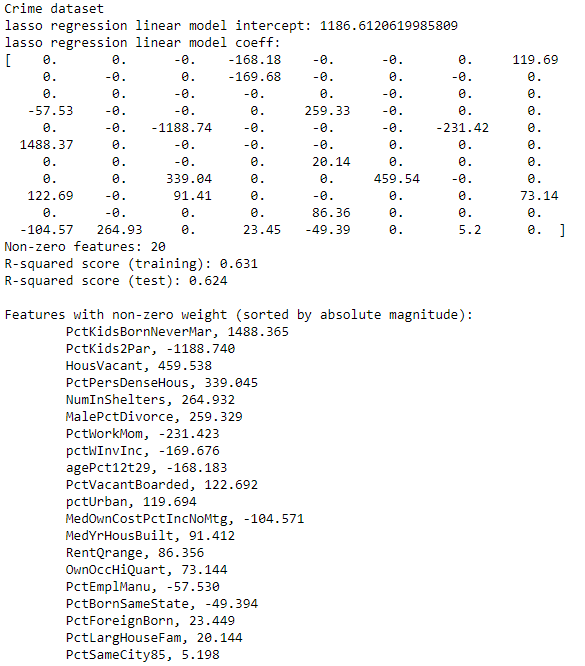
The ridge regression used L2 regularization, but Lasso uses L1 regularization. L1 adds in a regularization penalty that considers the model’s absolute gradient:



As we are trying to find the minimum value of RSS this L1 term works but reducing the number of features a training set will use as the abs(w) will drive unimportant values of w to zero. This is called a **spare solution** and is a kind of feature selection.

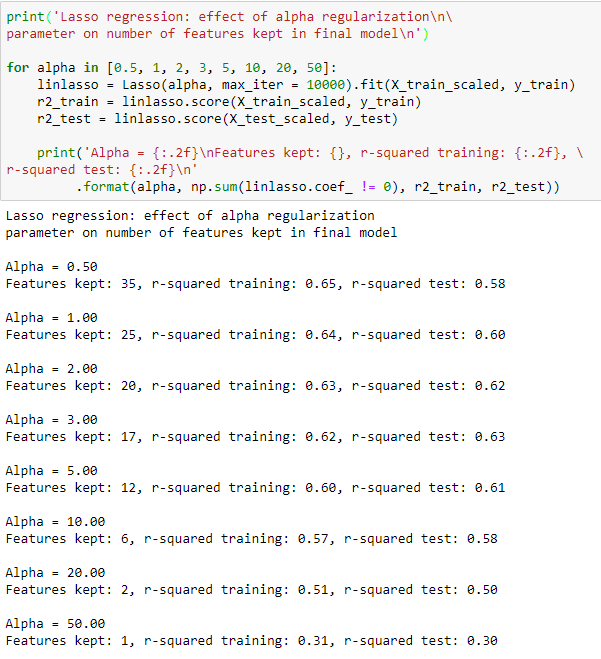
* Lasso is useful when there are many features and you only think a few are relevant.
* Ridge is useful if you think many of the variables effect the output.





From the above output we can see that the lasso outputs a list of sparse w coefficients meaning that the coefficients left are the most significant for the model’s accuracy with the training data. We can also see that the scores are pretty good compared to the other codes above!

**Tuning alpha for Lasso:**



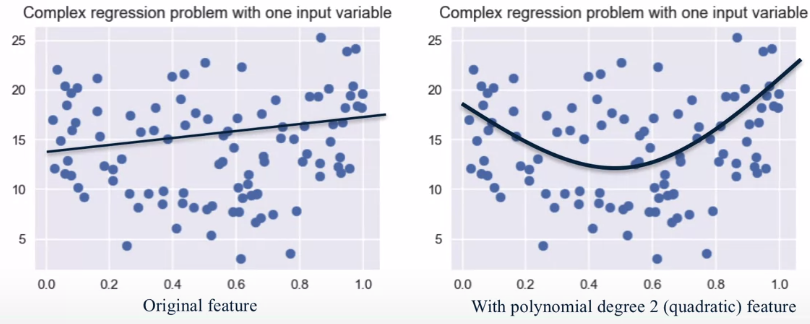
**Polynomial Regression:**

This is actually a transformation of the data into a feature space that is a polynomial. E.g. imagine we have a feature space of 2 variables, and we want to transform it into a 2n order polynomial:

And therefore:

This is still a linear model as the model is still based on a linear weighting of a combination of features, and we can use the same least squares estimator.

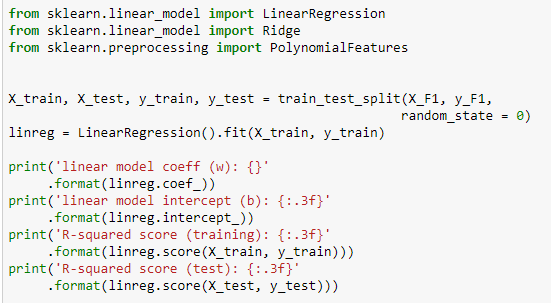
What this transformation does is it allows for polynomial models to be fit to the training data to allow use to reduce the bias of our models.

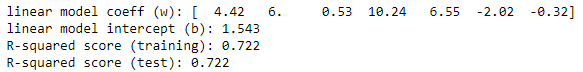


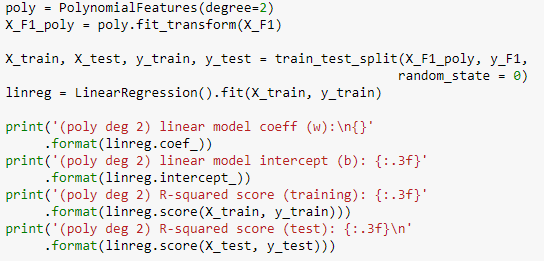
By carrying this transformation, we are considering the interaction between different features and seeing what effect they have on the model output. This can also help to make classification problems easier.

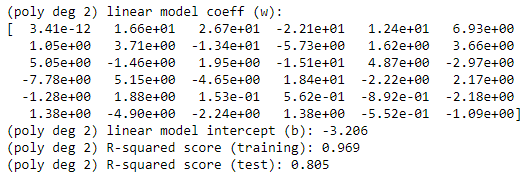
For example, it may be that housing prices vary as a quadratic function of both the lot size that a house sits on,

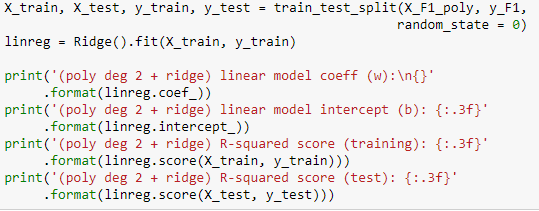
and the amount of taxes paid on the property.

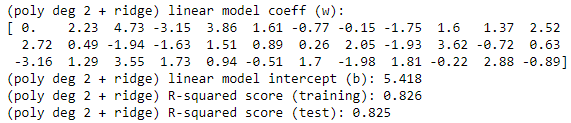












The model trained on the polynomial transformed data seems to perform the best.