Use housing price dataset. Using various parameters in the datasets we need to predict the housing price.

Dataset : Housing Price Prediction (in CSV format) 546 rows, 12 columns

	price	lotsize	bedrooms	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	prefarea
	42000.0	5850	3	1	2	yes	no	yes	no	no	1	no
	38500.0	4000	2	1	1	yes	no	no	no	no	0	no
	49500.0	3060	3	1	1	yes	no	no	no	no	0	no
	60500.0	6650	3	1	2	yes	yes	no	no	no	0	no
	61000.0	6360	2	1	1	yes	no	no	no	no	0	no

Preprocessing:

- All the columns containing the option as ‘yes’ and ‘no’ we remove them as we cannot use them in linear regression and we continue with the remaining data.
- Data splitted into features and prediction(X and Y, respectively) Prediction variable here is the ‘price’
- Data is normalized, and a Bias term added to feature set.

So our X has feature-set and Y is the house prices.
(1) X = m X (n + 1) matrix, containing the training samples features in the rows, where each x(i) is a (n + 1) X 1 matrix.
(2) m = number of training samples
(3) W(i) = the parameters
(4) x(0)= 1 for every sample.
(5) n = features count, not including x0 for every training sample
(6) Y contains all the m target values from the training samples, so it’s m X 1 matrix

Normal Equation method with regularisation

W = (X.T * X + L)^-1 * (X.T * Y)

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$

Extend input dimensions to add bias terms

In the cost function we add an additional “regularization” term in order to avoid overfitting of features. Overfitting happens when the model fits the training data more than required, and thus will lead to more error when we test it for a sample from the test data. The regularization term will thus add to the cost and will “reduce” the effect of the parameters will contribute towards overfitting.

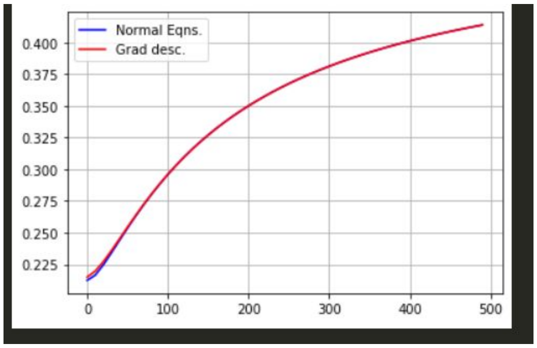
Analysis of Normal Equation with Regularization on the entire dataset:

Lambda	Final Cost
1	0.16376182676447748
100	0.24522819472882595
500	0.4111756460232463
1000	0.5330343556160958
1500	0.6406573247629825

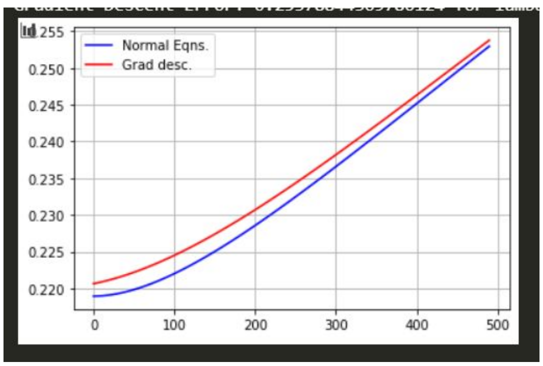
Gradient Descent Method with regularisation

By differentiating the cost function we get the result for gradient descent.This does depend on taking an appropriate value of λ though. If the value of λ is too high, then it reduces the effect of almost every feature, and thus it will lead to underfitting of the data. If it is too low, then it doesn’t have much effect and there is a chance that overfitting is still there.

The lamda vs cost graph taking 50 samples of the dataset



The lamda vs cost graph taking the entire dataset



Observations

We observe that when we don't split the dataset:

- Normal equations result is always better than Gradient descent
- Cost is always increasing with increasing lambda

The given dataset has very few features for overfitting to occur. Hence regularization doesn't play a major role here. Also it's possible to get a solution using GD better than normal equations when there's a train-test split