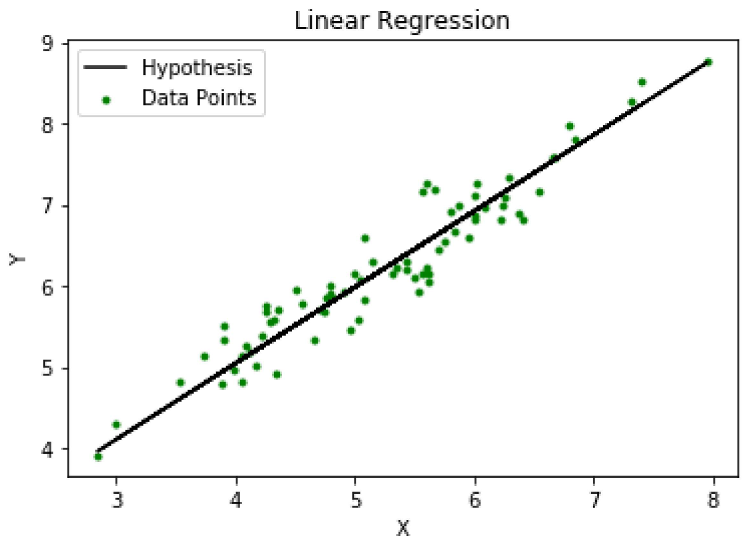
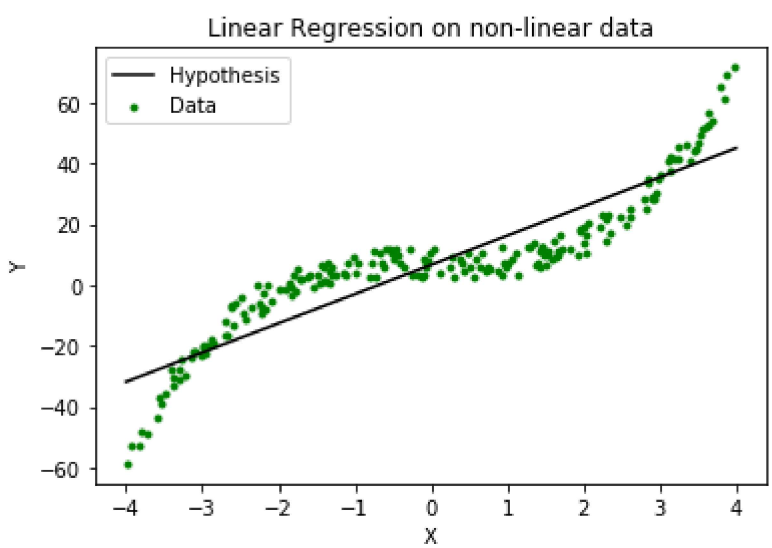
ML | Locally weighted Linear Regression

Linear Regression is a supervised learning algorithm used for computing linear relationships between input (X) and output (Y). The steps involved in ordinary linear regression are:

> **Training phase:** Compute $\theta$to minimize the cost. $J(\theta) = \sum_{i=1}^{m} (\theta^T x^{(i)} - y^{(i)})^2$
>
> **Predict output:** for given query point x, $return : \theta^T x$



As evident from the image below, this algorithm cannot be used for making predictions when there exists a non-linear relationship between X and Y. In such cases, locally weighted linear regression is used.



Locally Weighted Linear Regression:

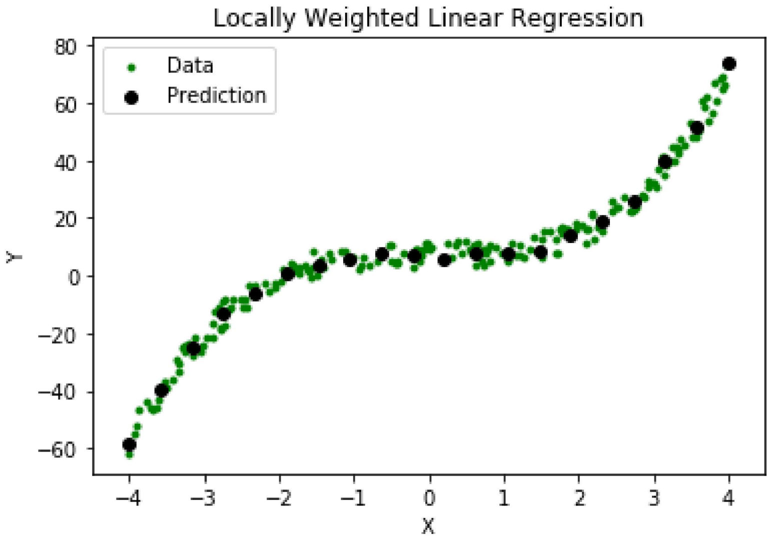
Locally weighted linear regression is a non-parametric algorithm, that is, the model does not learn a fixed set of parameters as is done in ordinary linear regression. Rather parameters $\theta$are computed individually for each query point x. While computing θ, a higher "preference" is given to the points in the training set lying in the vicinity of $x$than the points lying far away from x. The modified cost function is: $J(\theta) = \sum_{i=1}^{m} w^{(i)}(\theta^T x^{(i)} - y^{(i)})^2$ where, $w^{(i)}$is a non-negative "weight" associated with training point $x^{(i)}$. For $x^{(i)}$s lying closer to the query point x, the value of $w^{(i)}$is large, while for $x^{(i)}$s lying far away from $x$the value of $w^{(i)}$is small. A typical choice of $w^{(i)}$is: $w^{(i)} = exp(\frac{-(x^{(i)}-x)^2}{2\tau^2})$where $\tau$is called the bandwidth parameter and controls the rate at which $w^{(i)}$falls with distance from xClearly, if $|x^{(i)} - x|$is small $w^{(i)}$is close to 1 and if $|x^{(i)} - x|$is large $w^{(i)}$is close to 0. Thus, the training set points lying closer to the query point x contribute more to the cost $J(\theta)$than the points lying far away from x.

NOTE: For Locally Weighted Linear Regression, the data must always be available on the machine as it doesn't learn from the whole set of data in a single shot. Whereas, in Linear Regression, after training the model the training set can be erased from the machine as the model has already learned the required parameters.

For example: Consider a query point x= 5.0 and let $x^{(1)}$and $x^{(2)}$be two points in the training set such that $x^{(1)}$= 4.9 and $x^{(2)}$= 3.0. Using the formula $w^{(i)} = exp(\frac{-(x^{(i)}-x)^2}{2\tau^2})$with τ= 0.5: $w^{(1)} = exp(\frac{-(4.9-5.0)^2}{2(0.5)^2}) = 0.9802$ [Tex]w^{(2)} = exp(\frac{-(3.0 – 5.0)^2}{2(0.5)^2}) = 0.000335 [/Tex]

$$So,\ J(\theta) = 0.9802 * (\theta^T x^{(1)} - y^{(1)}) + 0.000335 * (\theta^T x^{(2)} - y^{(2)})$$

Thus, the weights fall exponentially as the distance between $x$and $x^{(i)}$increases and so does the contribution of error in prediction for $x^{(i)}$to the cost. Consequently, while computing θ, we focus more on reducing $(\theta^T x^{(i)} - y^{(i)})^2$ for the points lying closer to the query point (having a larger value of $w^{(i)}$).



Steps involved in locally weighted linear regression are:

> **Compute to minimize the cost.** $J(\theta) = \sum_{i=1}^{m} w^{(i)}(\theta^T x^{(i)} - y^{(i)})^2$
>
> **Predict Output:** for given query point x, $return : \theta^T x$

Points to remember:

- Locally weighted linear regression is a supervised learning algorithm.
- It is a non-parametric algorithm.
- There exists No training phase. All the work is done during the testing phase/while making predictions.
- The dataset must always be available for predictions.
- Locally weighted regression methods are a generalization of k-Nearest Neighbour.
- In Locally weighted regression an explicit local approximation is constructed from the target function for each query instance.
- The local approximation is based on the target function of the form like constant, linear, or quadratic functions localized kernel functions.