**Feature engineering and polynomial regression**

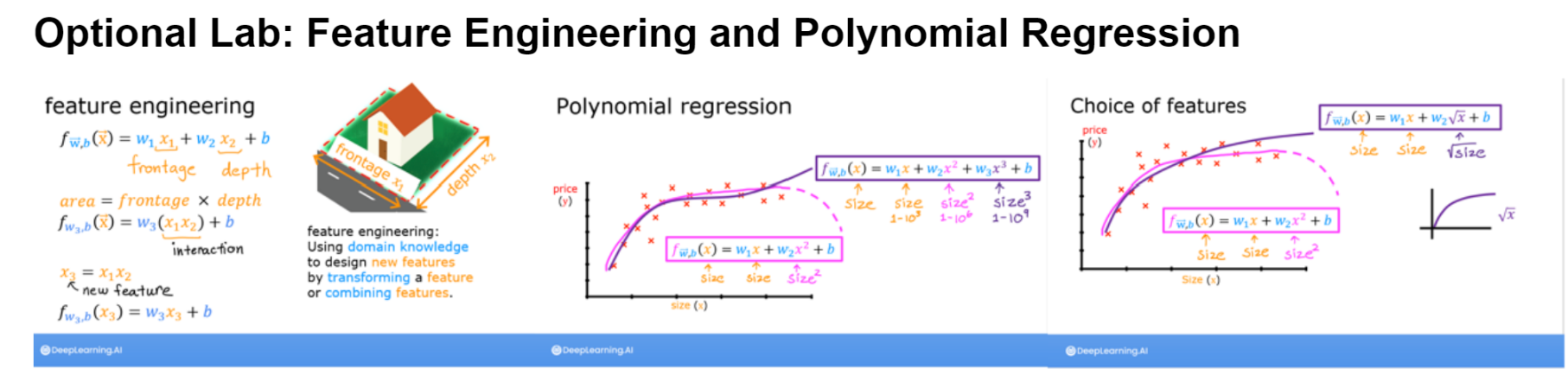


Figure 1 Feature engineering and polynomial regression basics

Polynomial regression is used when the input feature are combinations of features. Feature engineering is used to feat non-linear input data.

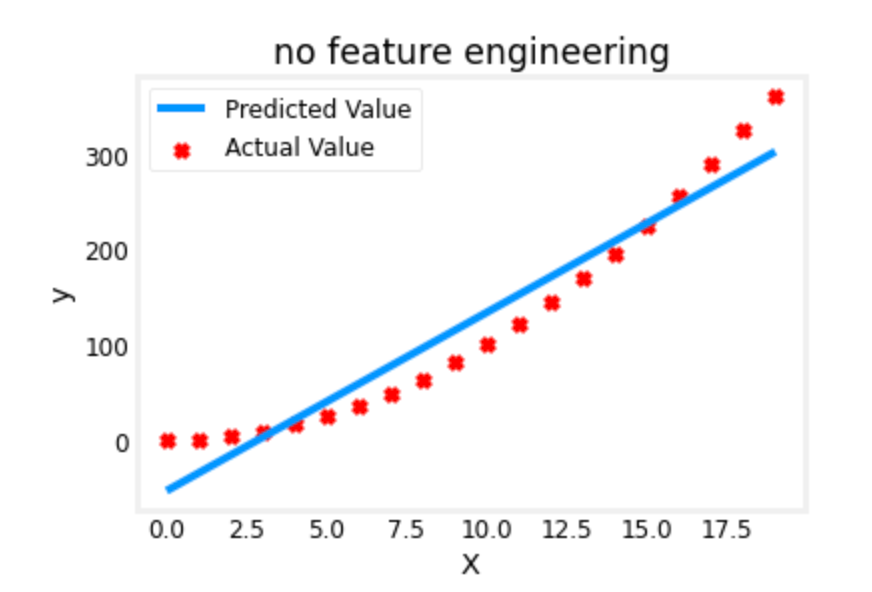


Figure 2 No feature engineering

Linear regression can’t fit a non-linear curve.

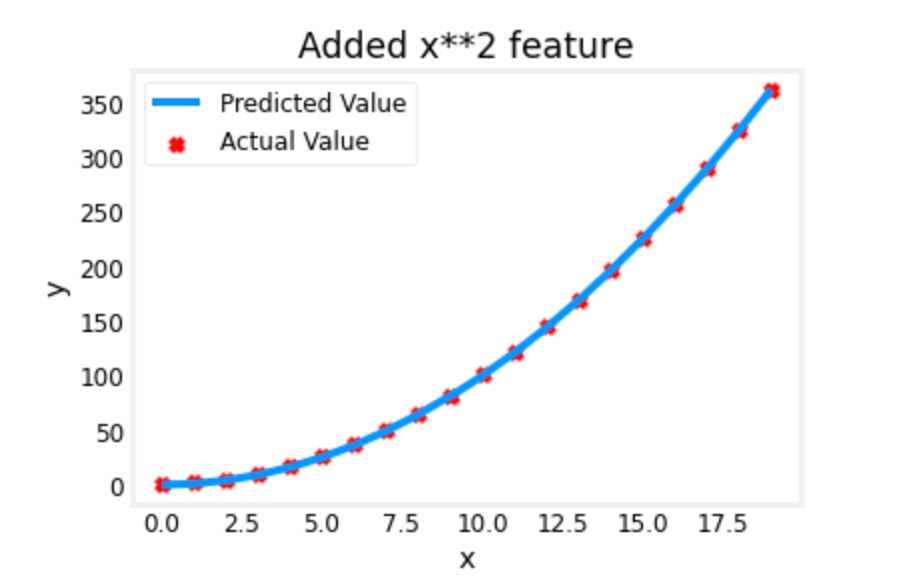
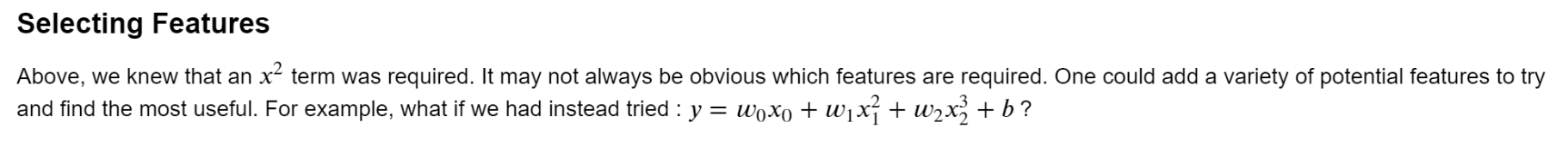
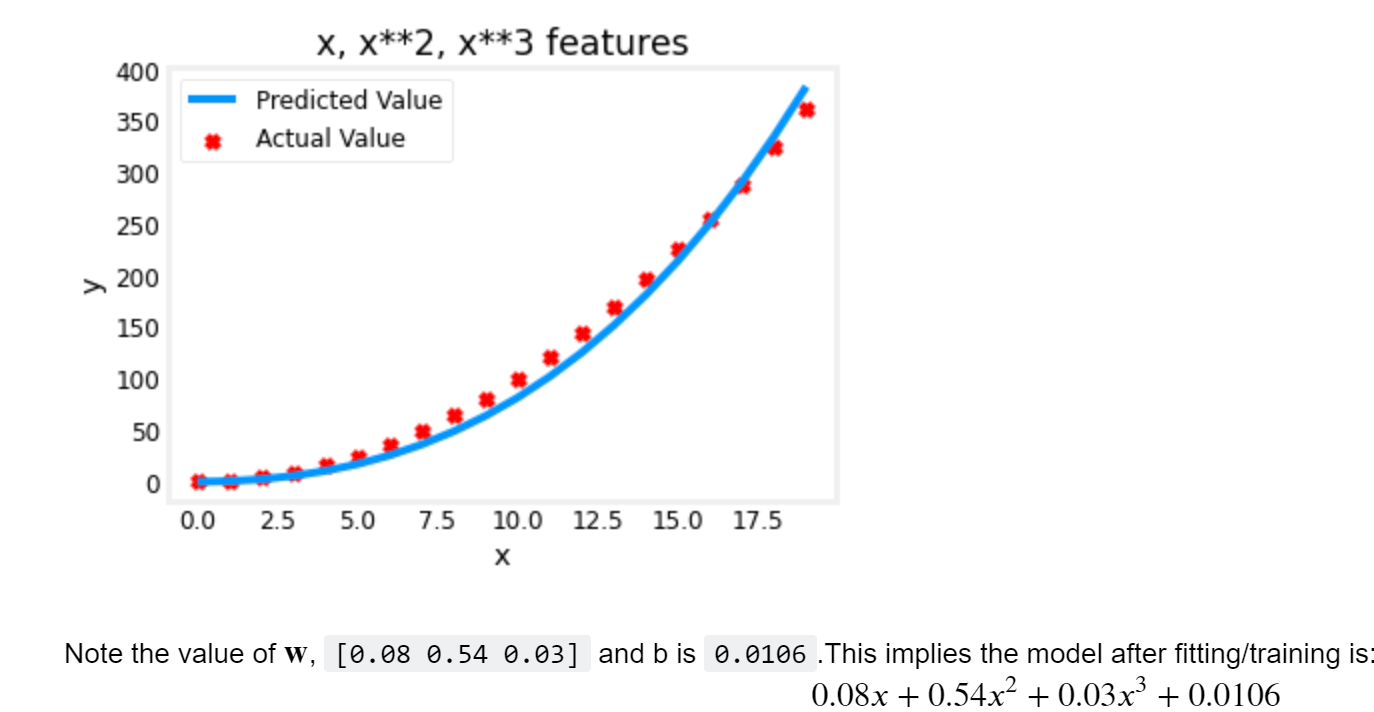


Figure 3 With x\*\*2

Much better fit of the parameters.





Gradient descent has emphasized the data that is the best fit to the x2 data by increasing the w1 term relative to the others. Gradient descent is picking the 'correct' features for us by emphasizing its associated parameter. So the less weight values indicate less important/correct features. When the weight become very small, the associated feature is not useful fitting the model to the data. Bigger the weight, most import is the associated feature.

**An Alternate View**

Above, polynomial features were chosen based on how well they matched the target data. Another way to think about this is to note that we are still using linear regression once we have created new features. Given that, the best features will be linear relative to the target. This is best understood with an example.

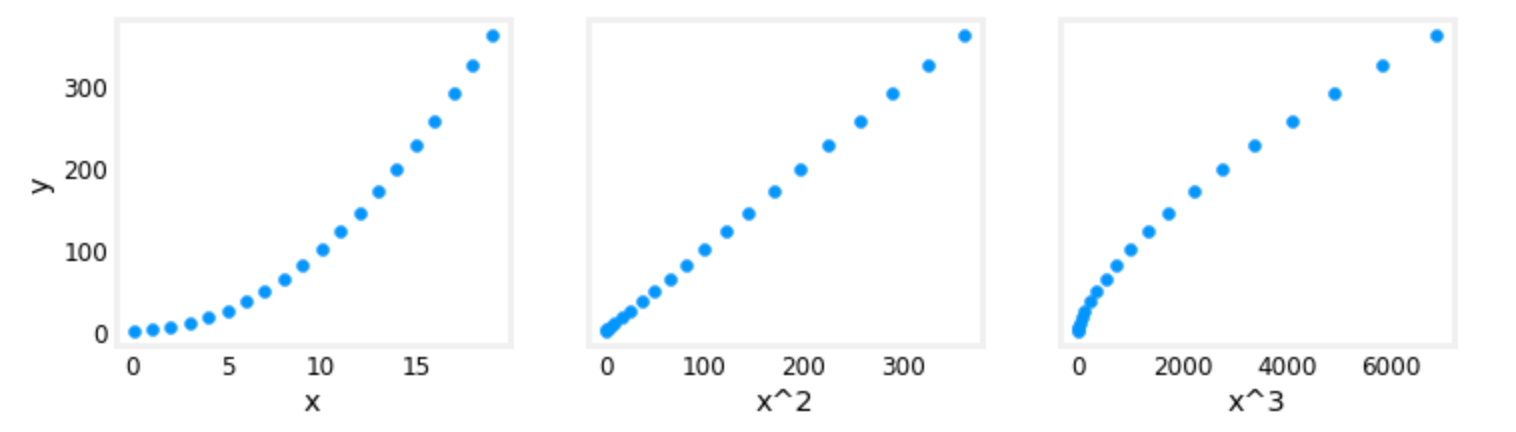


Figure 4 Feature mapping

Above, it is clear that the x^2 feature mapped against the target value y is linear. Linear regression can then easily generate a model using that feature.

**Scaling features**

As described in the last lab, if the data set has features with significantly different scales, one should apply feature scaling to speed gradient descent. In the example above, there is 𝑥x, 𝑥2x2 and 𝑥3x3 which will naturally have very different scales. Let's apply Z-score normalization to our example.

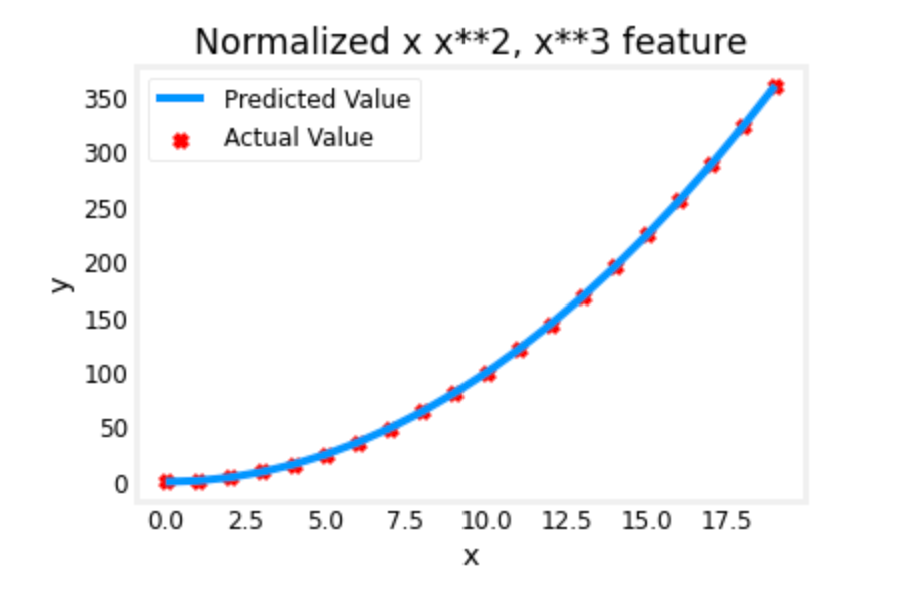


Figure 5 Feature engineering with normalization

We use the same number of iterations as with the unnormalized data. Feature scaling allow a faster convergence, thus a more performant model.

w,b found by gradient descent: w: [5.27e-05 1.13e+02 8.43e-05], b: 123.5000

Notice that the term w1 and w3 are very small compared to w2.

**Complex Functions**

With feature engineering, even quite complex functions can be modeled:

