**Polynomial Regression**

Table of Contents

[Modifying Features 2](#_Toc100149697)

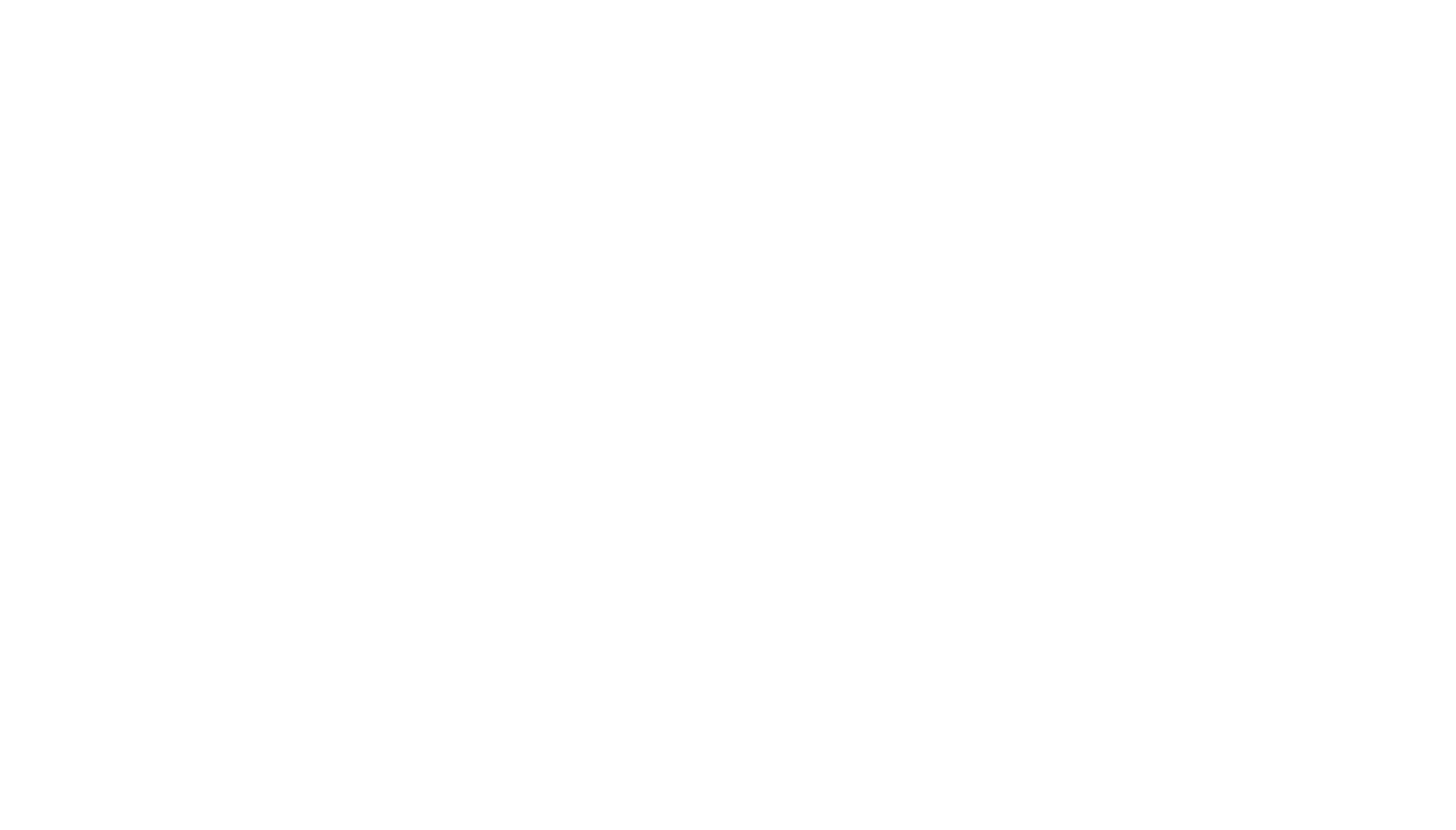
[Overfitting 3](#_Toc100149698)

There are situations where linear regression does not give us all that good a fit, where a **polynomial** equation would probably have done the job better. In cases like those, we can use **polynomial regression**.

## Modifying Features

Consider that we have a hypothesis , where is the height and is the width of some shape. This is a linear equation. However, if we feel that a polynomial equation would be a better fit for the data, we can turn this into a polynomial function by **modifying the features**. For example, we can create a new parameter , the area of the shape. Thus, .

Even if we have just one feature, we can use that in different ways to create a **polynomial function**. For example, if is our only feature, we can still use .



**Quadratic equations** like this tend to bend back downwards as we keep increasing . If this is undesirable (e.g. with housing prices, which should not decrease), we can use a **cubic** function instead, i.e. . Alternatively, we can use an equation like , which also does not decrease. Really, the choice of the function is up to us. It can be whatever we feel will fit the data well.

However, note that when we modify features in this way, **feature scaling** is essentially. The first feature, , and the second feature, , can have wildly different ranges, so we have to scale down both features accordingly.

## Overfitting

If using a polynomial function gives us a better fit for the data, we should keep going shouldn’t we? If is a good fit, surely will be an even better fit. While this is entirely true, it really will fit the data well, we will eventually run into the problem of **overfitting**.

Consider that we manage to somehow create a hypothesis that perfectly fits all our data points. There is no error at all. However, this was just the training data. The test data and real-world data will be different. There is no guarantee that our hypothesis will fit that data well. If we have a hypothesis that fits the training data really well but has a huge error for the test data, we have **overfitted** the hypothesis. The hypothesis has essentially been created especially for the training data and won’t work with other data. This is not what we want.

So, what do we do if we just happen to have a hypothesis that has been overfitted. There may be cases where we have a **huge number of features** that causes this problem to occur. In that case, we have to **remove** some features. It is possible to do this manually, for example by removing redundant features, but this is difficult to do. A better method is to use **regularization**. We will look into the details of how regularization works later on, but in summary, it sets the value of if we want to remove the feature .