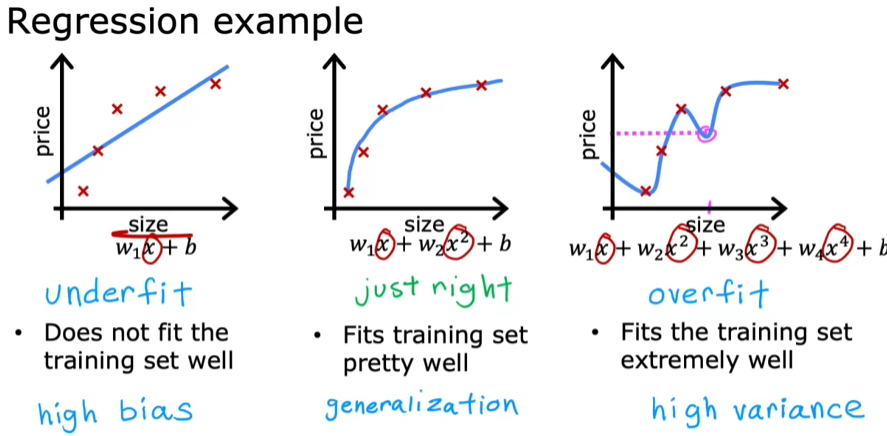
**THE PROBLEM OF OVERFITTING**

**Understanding Overfitting and Underfitting**

* **Overfitting occurs when a model learns the training data too well, capturing noise and fluctuations, which leads to poor generalization on new data. This is often associated with high variance.**
* **Underfitting happens when a model is too simple to capture the underlying patterns in the data, resulting in poor performance on both training and new data. This is linked to high bias.**

**Examples of Model Fitting**

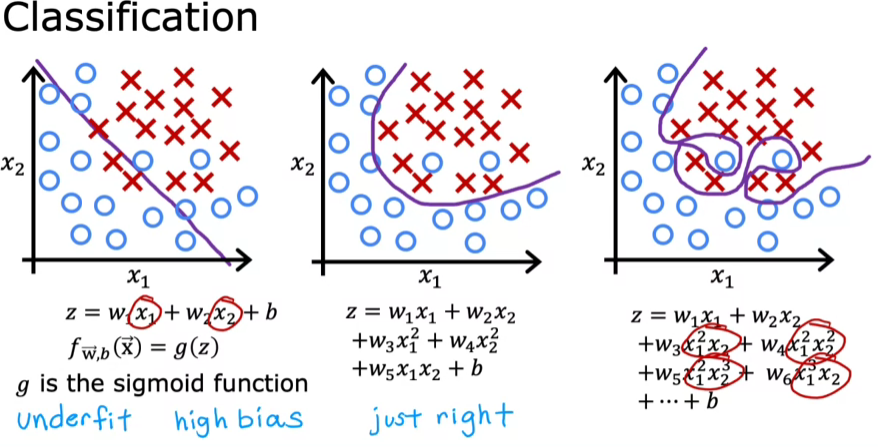
****

* **A linear regression model may underfit the data if it assumes a linear relationship when the actual relationship is more complex, leading to high bias.**
* **A high-order polynomial model can overfit the data by fitting every training example perfectly, resulting in a wiggly curve that does not generalize well to unseen data.**

**Finding the Right Balance**

* **The goal in machine learning is to find a model that is "just right," meaning it neither underfits nor overfits the data, achieving a balance between bias and variance.**
* **This balance can be likened to the story of Goldilocks, where the ideal model is one that captures the essential patterns without being overly complex or too simplistic.**

**This applies not only to regression but to classification as well.**

****

**ADDRESSING OVERFITTING**

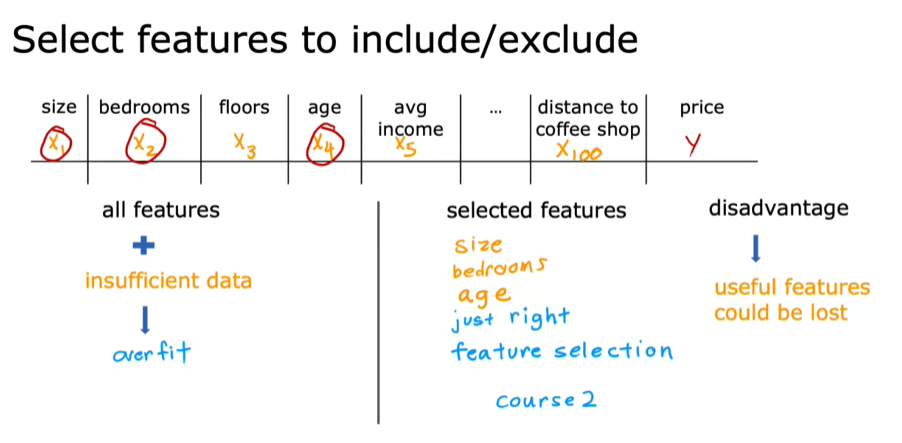
**Collecting More Data**

* **One effective way to combat overfitting is to gather more training data, which helps the learning algorithm create a less complex model.**
* **If more data is available, it allows the algorithm to learn from a broader range of examples, improving its generalization.**

****

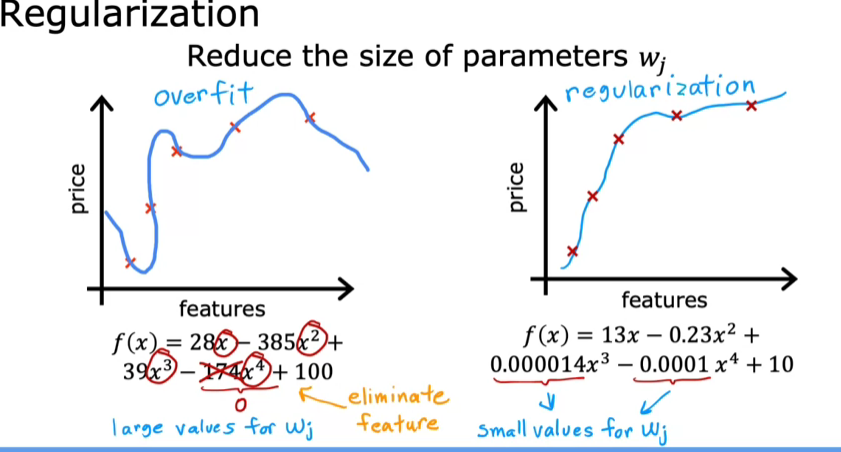
**Feature Selection**

1. **Reducing the number of features used in the model can also help mitigate overfitting. For instance, selecting only the most relevant features can lead to better performance.**
2. **While feature selection can simplify the model, it may also discard useful information, so careful consideration is needed.**

****

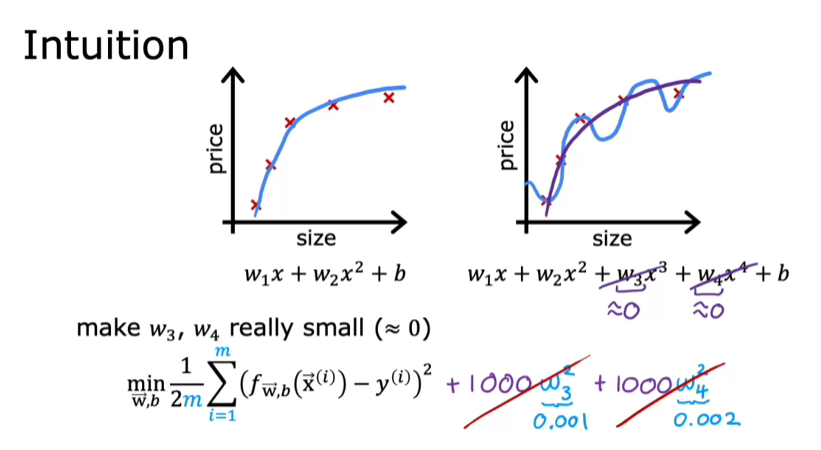
**Regularization Techniques**

* **Regularization is a method that reduces the impact of certain features without completely removing them, encouraging smaller parameter values.**
* **This technique helps maintain all features while preventing any single feature from dominating the model, thus improving its fit to the training data.**

****

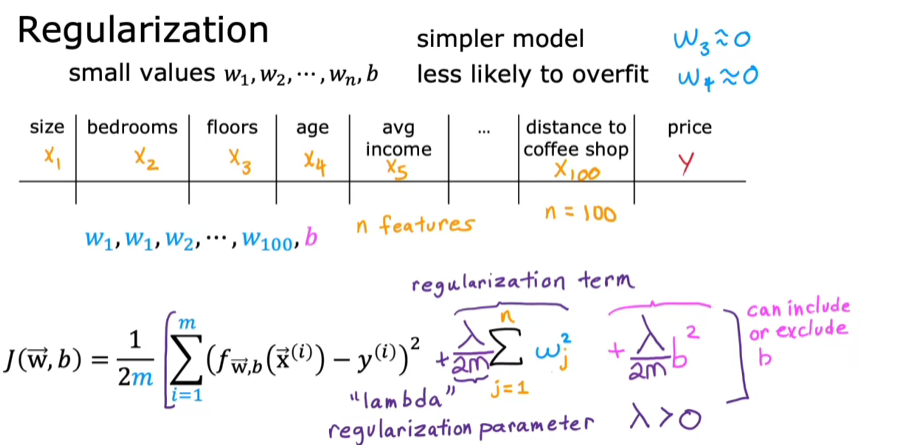
**COST FUNCTION WITH REGULARIZATION**

**Understanding Regularization**

****

* **Regularization aims to keep the parameter values (W1 through WN) small, which helps in creating a simpler model that is less prone to overfitting.**
* **By adding a penalty term to the cost function, such as 1000 times W3 squared plus 1000 times W4 squared, we encourage smaller values for these parameters.**

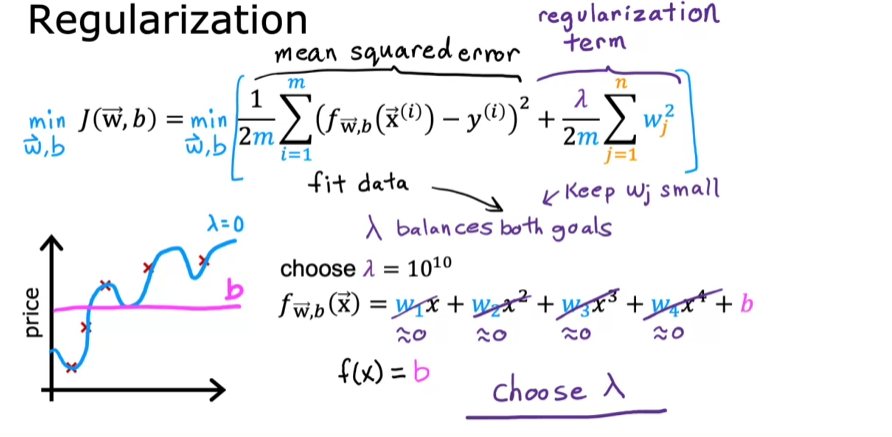
**Modified Cost Function**

****

* **The modified cost function includes the original mean squared error cost plus a regularization term, which penalizes all parameters (W1 to W100) to keep them small.**
* **The regularization parameter, lambda (λ), determines the trade-off between fitting the training data well and keeping the parameters small.**

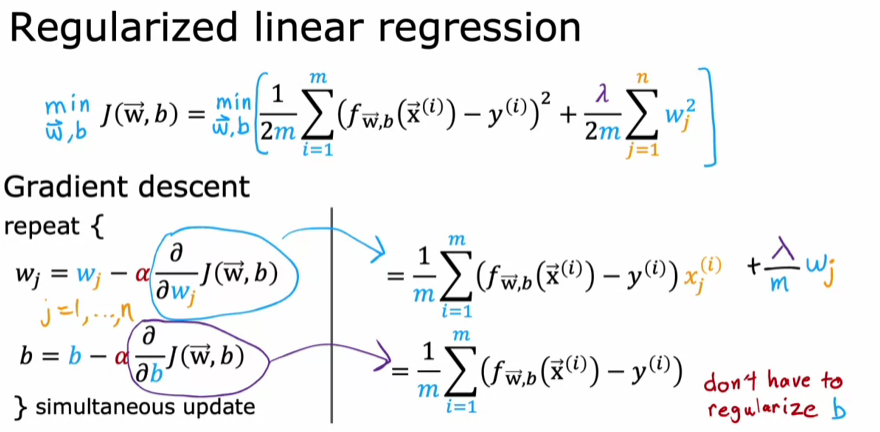
**Effects of Lambda**

* **If λ is set to 0, the model may overfit the data, resulting in a complex curve.**
* **Conversely, if λ is set to a very large value, the model may underfit, leading to a simple horizontal line. The goal is to find a balanced λ that minimizes both the mean squared error and the parameter values effectively.**

****

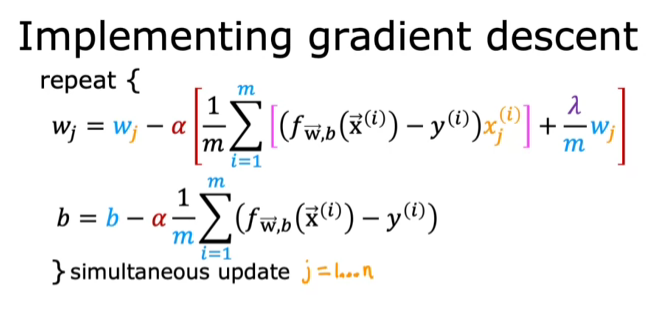
**REGULARIZED LINEAR REGRESSION**

* **The cost function for regularized linear regression includes the usual squared error term and an additional regularization term, where Lambda is the regularization parameter.**
* **The goal is to minimize this regularized cost function by updating parameters w and b using gradient descent.**

****

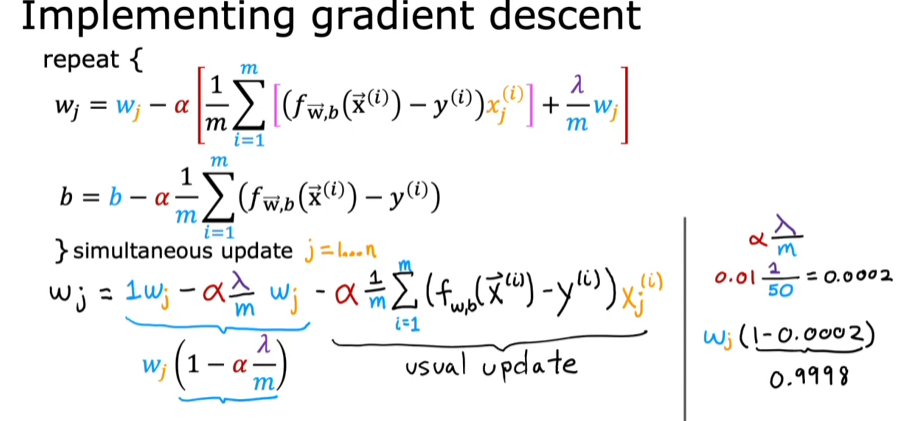
**Gradient descent updates**

* **The update rules for parameters w and b remain similar to those in unregularized linear regression, with the key difference being the additional term in the derivative of the cost function with respect to wj.**
* **The update for wj includes a term that shrinks wj slightly on each iteration, effectively reducing overfitting.**

****

**Intuition behind regularization**

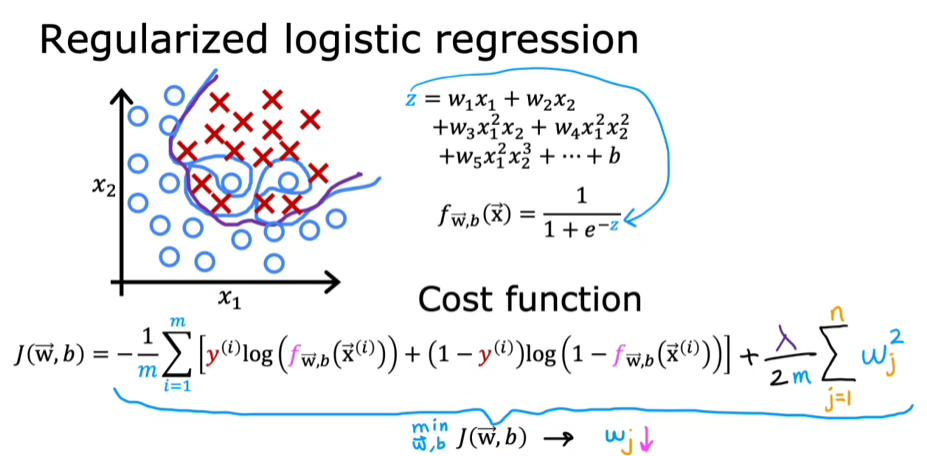
* **Regularization works by multiplying wj by a number slightly less than 1, which gradually shrinks the values of wj during each iteration of gradient descent.**
* **This approach helps improve the performance of linear regression, especially when dealing with many features and a small training set.**

****

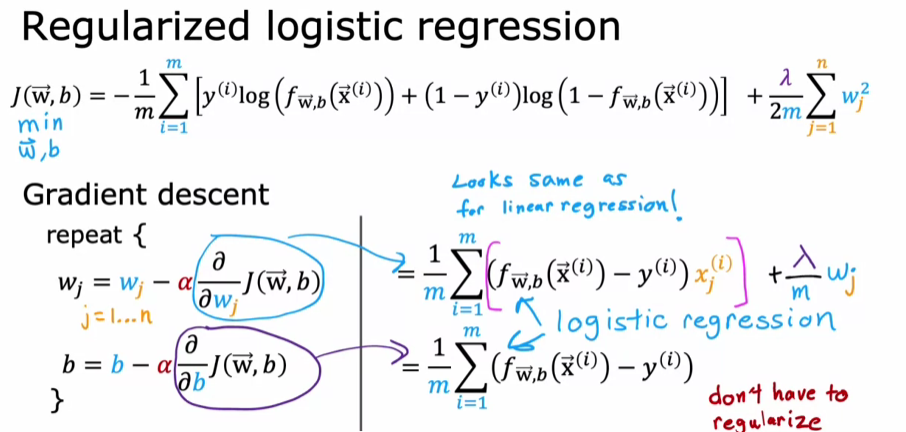
**REGULARIZED LOGISTIC REGRESSION**

**Understanding Overfitting in Logistic Regression**

* **Logistic regression can overfit when using high-order polynomial features, leading to overly complex decision boundaries that do not generalize well to new data.**
* **Regularization helps mitigate overfitting by adding a penalty term to the cost function, which discourages large parameter values.**

****

**Implementing Regularized Logistic Regression**

****

**Key Takeaways**

* **Regularization is crucial for improving model generalization, especially when dealing with many features.**
* **Understanding and applying concepts like cost functions and gradient descent are foundational skills that can lead to valuable applications in machine learning.**