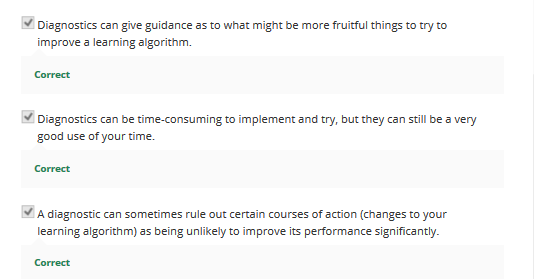
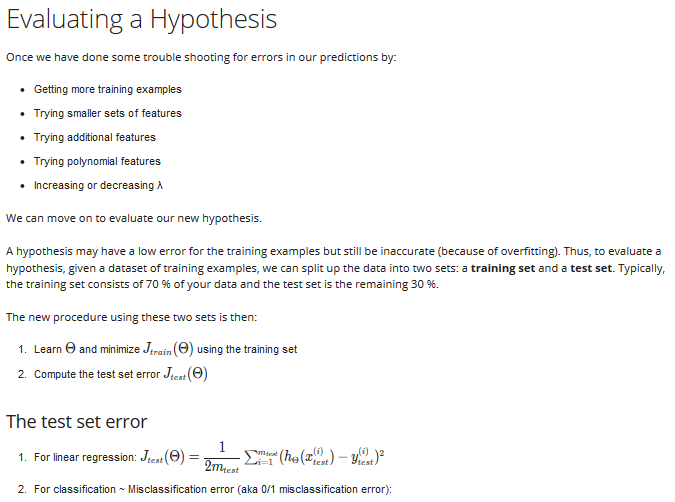
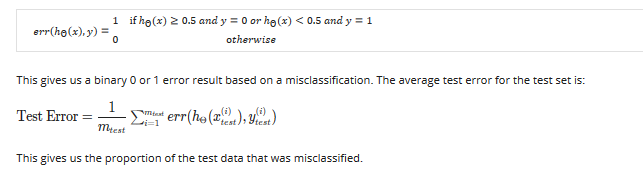
**Diagnostics of Machine Learning algorithm:**

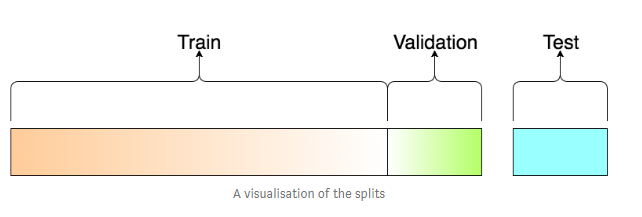
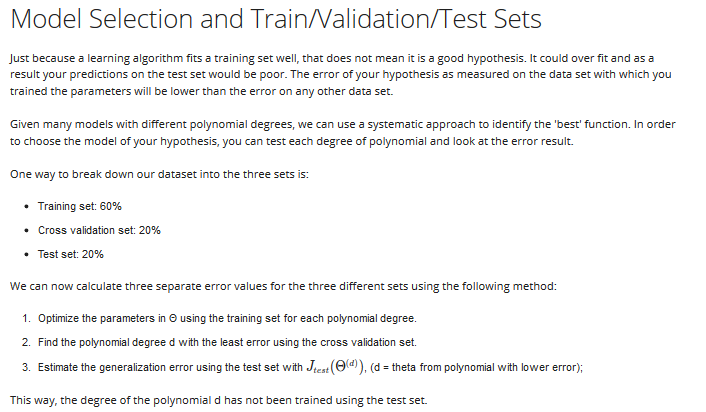
Summary:

* **Getting more training examples:** Fixes high variance
* **Trying smaller sets of features:** Fixes high variance
* **Adding features:** Fixes high bias
* **Adding polynomial features:** Fixes high bias
* **Decreasing λ(Lambda):** Fixes high bias
* **Increasing λ(Lambda):** Fixes high variance.



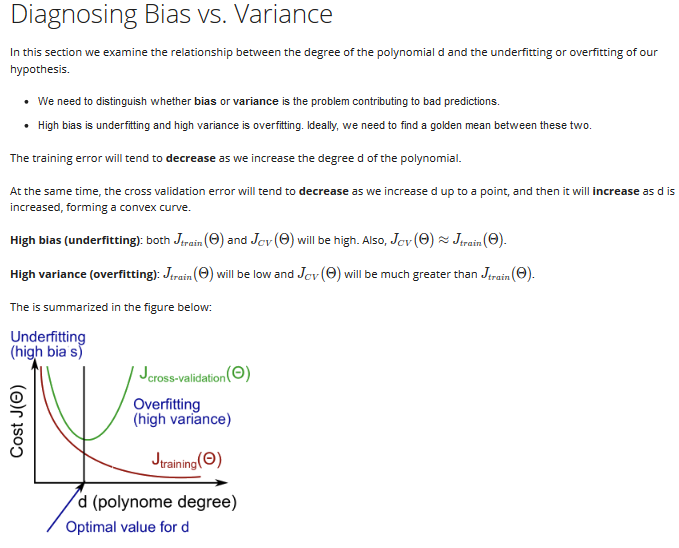
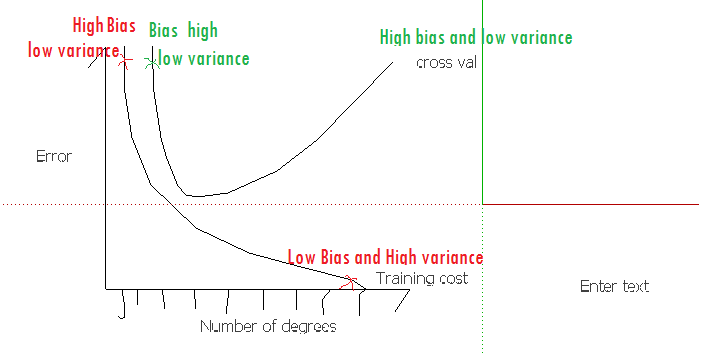


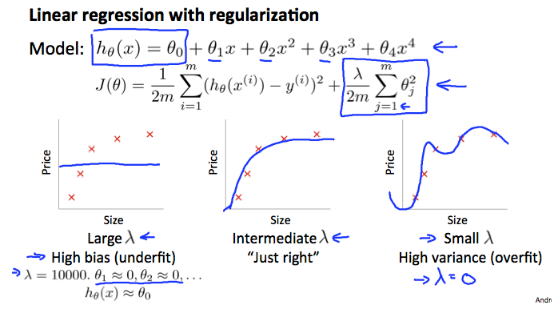


Train\_x, train\_y,**Val\_x,Val\_y**= train\_test\_split(trainingdata)

**Training Error:** We get the by calculating the classification error of a model on *the same data the model was trained on*.

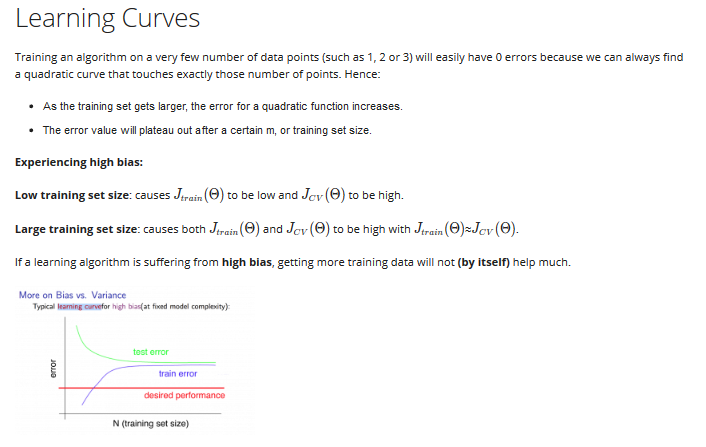
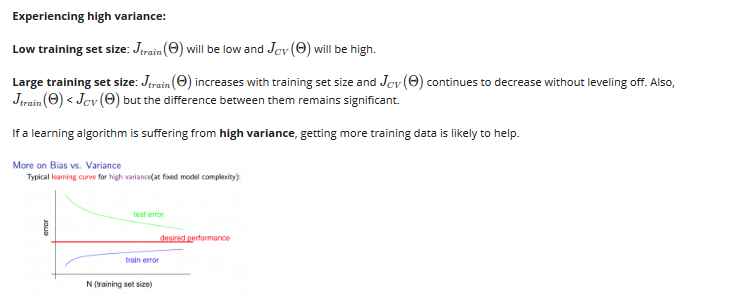


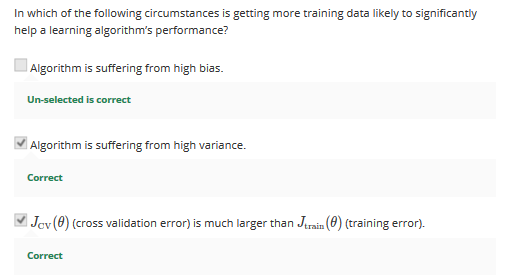


If we have high lambda value then, Underfit

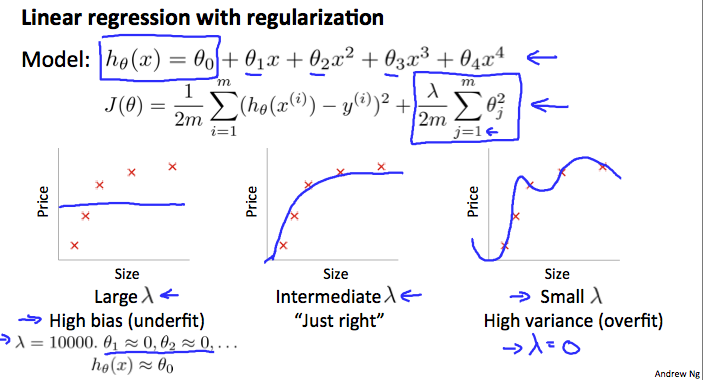
If we have small lambda value then , Overfit

Then we need to set a sweet inbetween value.





Lambda and its effects:



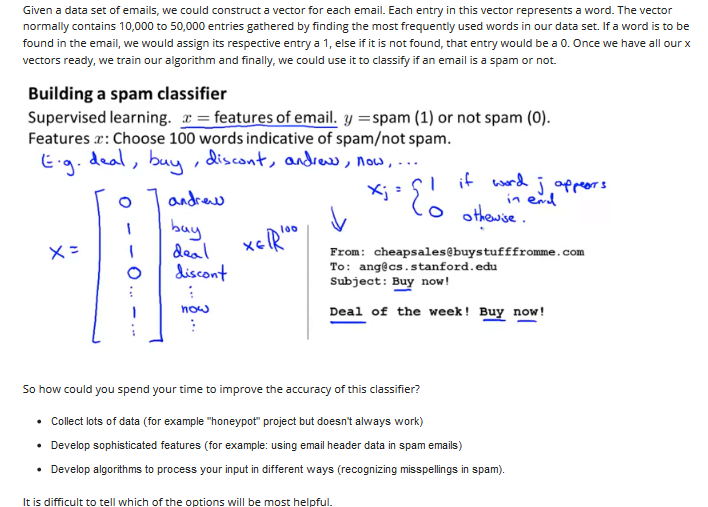
Underfit means, High Bias and reducing lambda will help us in Fixing Lambda

High Lambda, will penalize each theta and will make them close to zero

Our decision process can be broken down as follows:

* **Getting more training examples:** Fixes high variance
* **Trying smaller sets of features:** Fixes high variance
* **Adding features:** Fixes high bias
* **Adding polynomial features:** Fixes high bias
* **Decreasing λ:** Fixes high bias
* **Increasing λ:** Fixes high variance.

Building Spam Classifier:



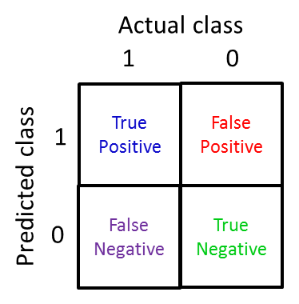
Q: Why it is recommended to perform error analysis using the cross validation data rather than the test data ?

Answer: If we develop new features by examining the test set, then we may end up choosing features that work well specifically for the test set, so Jtest(θ) is no longer a good estimate of how well we generalize to new examples.

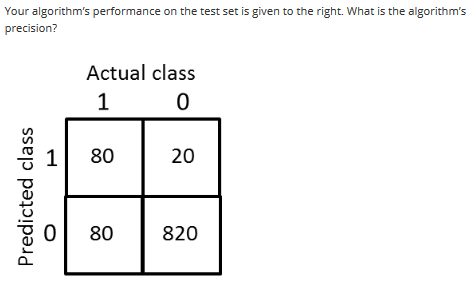
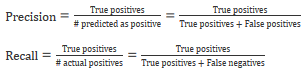
How to Evaluate Metrics?

Using:

Precision/Recall:



Recall : 0.8



Precision: 0.8