Machine Learning Diagonistic:

Diagonistic is a test that you can run to gain insight what is/isn't working with a learning algorithm, and gain guidance as to how best to improve its performance.

Troubleshooting for errors in our prediction by

- → Getting more training examples
- → Trying smaller set of features
- → Trying additional features
- → Trying polynomial features
- → Increasing or Decreasing λ

To evaluate a hypothesis we can split our dataset into two parts → 70% → Training set
30% → Test set

The test set error:

$$J_{test}(\theta) = \frac{1}{2\, m_{test}} \sum_{i=1}^{m_{test}} \left(h_\theta\left(x_{test}^{(i)}\right) - y_{test}^{(i)} \right)^2$$

For 0/1 classification error →

$$err\left(h_\theta(x) - y\right) = \begin{cases} 1, & \text{if } h_\theta(x) \geq 0.5 \text{ and } y = 0 \\ & \text{or, } h_\theta(x) < 0.5 \text{ and } y = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{TEST ERROR} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err\left(h_\theta\left(x_{test}^{(i)}\right) - y_{test}^{(i)}\right)$$

Another way to divide out data set -

→ Training set: 60%

→ Cross Validation set: 20%

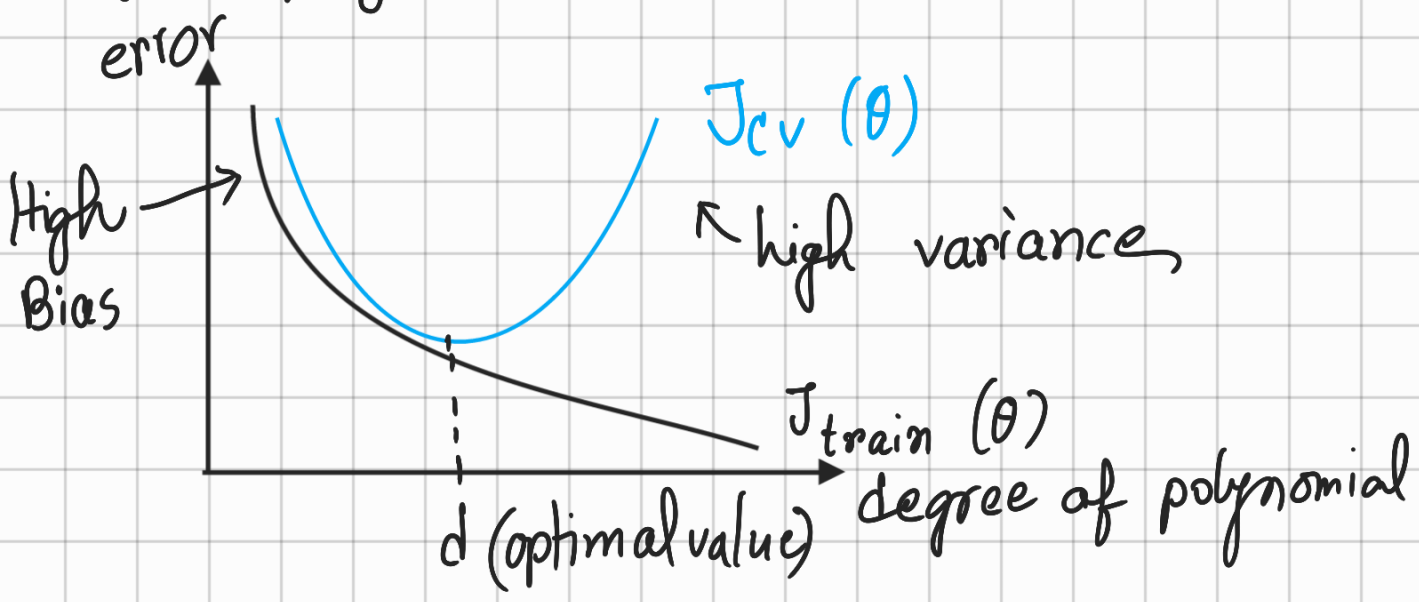
→ Test set: 20%

Now calculcate error and

→ Optimize the parameters in Θ using the training set for each polynomial degree.

→ Find the polynomial degree d with the least error using cross validation set

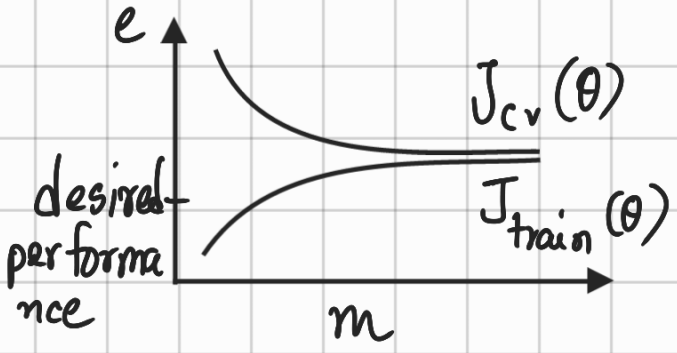
→ Estimate the generalization error using the test set with $J_{test}(\Theta^{d})$ d = theta from d degree polynomial with lower error.



Learning Curves: training set (m) vs Error (e)

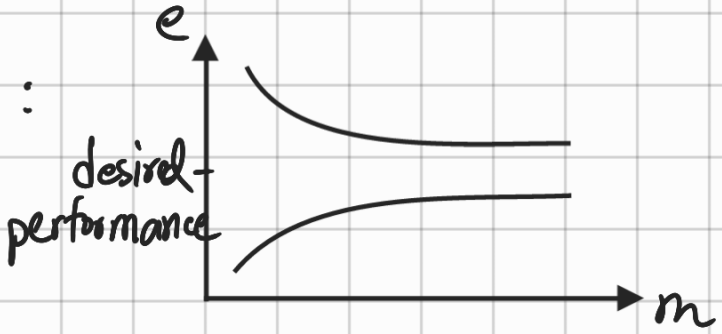
High Bias:

m ↑ will not help



High Variance:

m ↑ improve



ML Algorithm debugging:

	Fixes High
→ Getting more training Examples	→ Variances
→ Trying smaller set of features	→ Variances
→ Adding features	→ Bias
→ Adding polynomial features	→ Bias
→ Decreasing λ	→ Bias
→ Increasing λ	→ Variance

Building a spam classifier

- → Collect lots of data (still not sure wheather work or not)
- → Develop sophisticated feature (using email header data)
- → Develop algorithm to process input in different ways (recognizing misspellings)

"Stemming software" → universe = university

We have to do error analysis

→ Start with a simple algorithm, implement it quickly and test it early on cross validation data

→ Plot learning curve to decide if more data, features are likely to help.

→ Manually check the errors on the examples in the cross validation set and try to spot a trend where most of the error were made.

[We've to try & check many things, whichever decrease my error we're to pick it.]

Skewed data: If more than half error is undetected.

Precision: $\frac{TP}{TP+FP}$

Recal: $\frac{TP}{TP+FN}$

		Actual class 1	Actual class 0
Predicted class	1	TP	FP
	0	FN	TN

F-1 Score: $2\frac{PR}{P+R}$

In machine Learning

"It's not who has the best algorithm that wins it's who has the most data".

→ We have to have enough information to predict y.

→ The more data we can collect

→ With large dataset & small features we can have best output with a logistic regression

→ If we have large set of features than we have to use neural network.