**Machine Learning : Limits of Learning**

**Bayes optimal classifier**

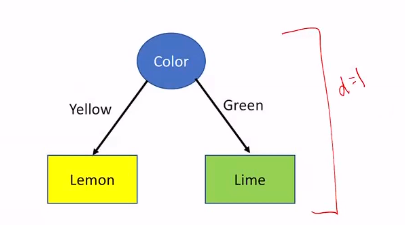
* Distribution D over
  + achieves minimal 0/1 error of any classifier.
    - **Proof by contradiction :** Suppose classifier g is better than
      * There must exist some datapoint, where they give you different labels or else they would have the same performance, the same error.
      * Bayes error rate is best possible error rate.
      * Probability that is wrong is
      * Probability that g is wrong is
      * , minimize error.
      * Probability of error for
      * Therefore is optimal. Q.E.D
  + Bayes theorem is the best, but we don’t use this method because we don’t have knowledge of the distribution we only guess the distribution based on a very limited sample rather than the distribution itself.
  + So, when the learning algorithms address this problem in different ways with different assumptions they’ll have different biases.

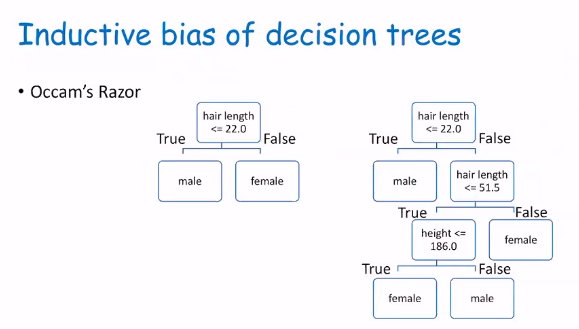
**Inductive Bias**

* Preference for a particular type of concept

**Decision Stump**

* If you only query one variable and create a tree out of it, it’s called a decision stump.





* **Occam’s Razor :** If there are multiple theories that are consistent with the data, the smallest is usually the best.
* So, the inductive bias of decision trees is to prefer smaller trees.
* **Representation bias:** Cannot easily represent **parity** (oddness or evenness) with decision tree.
* Due to learning algorithm not the data

**Not Everything is Learnable**

* **Things that can be fixed**
  + Noise in the data/label
  + Feature space is insufficient.
  + Some instances have more than one label.
  + Bias
    - Inductive Bias
    - Representation Bias

**No Free Lunch Theorem**

* Wolpert and Macready, 2005
* Any inductive bias that we choose will equal accuracy compared to any other bias when it’s average over all possible problems. Assuming they are all equally likely. So, if a bias is correct on some problems it could be incorrect on other problems.

**Underfitting**

* Opportunity to learn something but didn’t.
  + Indicates that our concept is not complex enough to model the complexity of the problem.

**Overfitting**

* Fit too tightly to minor details of training data.
  + Too many details that do not generalize well to the training data.
  + On training data error is 0% but on testing data it’s more 50%
  + Goal is not to get 0% on training data (that’s easy) but to minimal error on future data we haven’t seen yet.

**Random Guess**

* Balanced Data (50% positive, 50% negative)
  + Random guess will give you 50% error.
* So, if you have better accuracy and lower error than random guess that’s a good indication.

**Selecting Training and Test Data**

* Use a portion of data for training and a portion for testing.
* **Central Limit Theorem :** If the sample is large enough, 80/20 is a good split.
  + 80% for training, 20% for testing.
  + 70/30, 90/10 are also possible distribution.
* Don’t look at the test data before you design the algorithm because it’ll bias your method and results.

**Practical Considerations**

* Model parameters and hyperparameters
  + Hyperparameters are not manipulated by the learning algorithm, decided ahead of time when you design the algorithm.
* Like for decision tree a hyperparameter would how deep you would let the tree go.
  + If decision tree has a depth of 0 it’s just a majority classifier and will underfit.
  + If it’s allowed to go as deep as possible, it’ll have a long tree and can overfit.
  + So, designers might cut it off to promote generalization.
  + If they do decide to cut it off and prune that’s a hyperparameter.
* **Tree-way split.**
  + Training data 70%
  + Development data (Validation data) 10%
  + Test data 20%