**KNN:**

* k-NN classifiers are (called) lazy learners since they do not build models explicitly
* Classifying unknown records is relatively expensive
* Can produce arbitrarily shaped decision boundaries
* Easy to handle variable interactions since the decisions are based on local information
* Selection of right k is essential, (large k = tends to overfit, small k = sucks with noise)
* Superfluous or redundant attributes can create problems
* Missing attributes are hard to handle

**Rule based:**

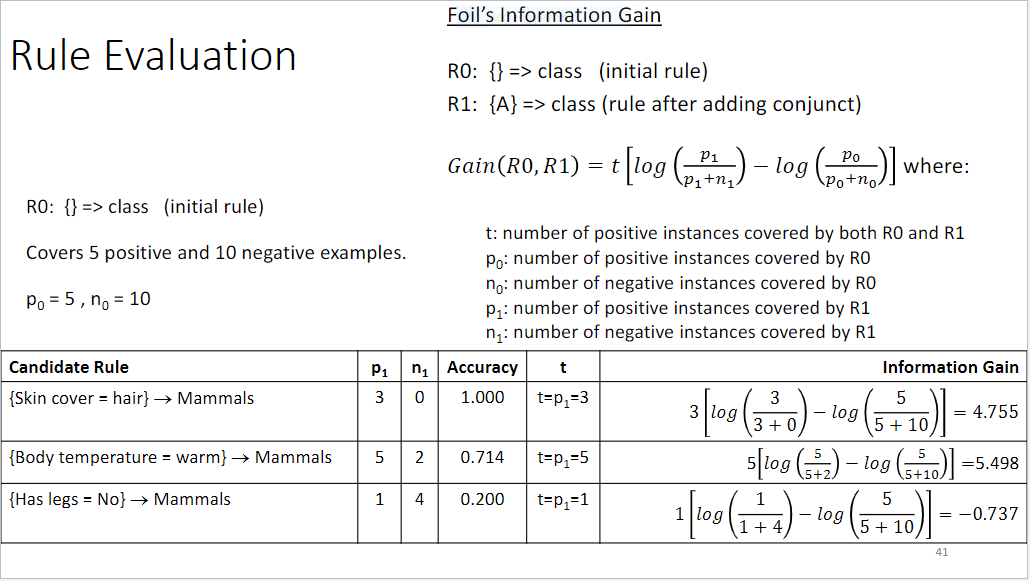
* Better suited for handling imbalanced classes
* Harder to handle missing values in the test set

They have characteristics similar to decision trees:

* As highly expressive as decision trees
* Easy to interpret
* Check straight lines probabilities, if not conflict, not good
* Performance comparable to decision trees
* Can handle redundant attributes

**Foil:**

The first candidate (expansion) rule has the highest accuracy of the three candidates. But the second candidate (expansion) rule has the highest information gain because of increased coverage. So, the second candidate rule is chosen.



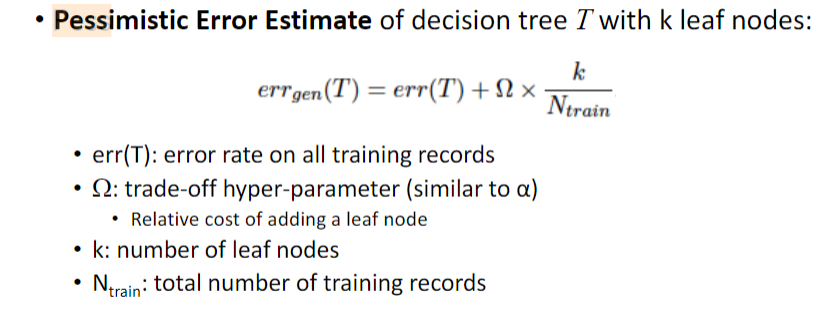
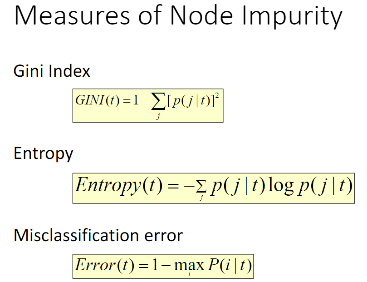
**Naïve Bayes:**

* Robust to isolated noise points
* Handles missing values by ignoring the instance during probability estimate calculations
* Robust to irrelevant attributes
* Check straight lines probabilities, if not conflict, not good
* Independence assumption may not hold for some attributes–Use other techniques such as Bayesian Belief Networks (BBN)

**Decision Trees:**

* Good for everything.
* Numbers of possible trees is exponentially large.

Purity measures and pessimistic error rate:



**Random Forests:**

* Votes among trees to find the majority vote for class. (no weights)

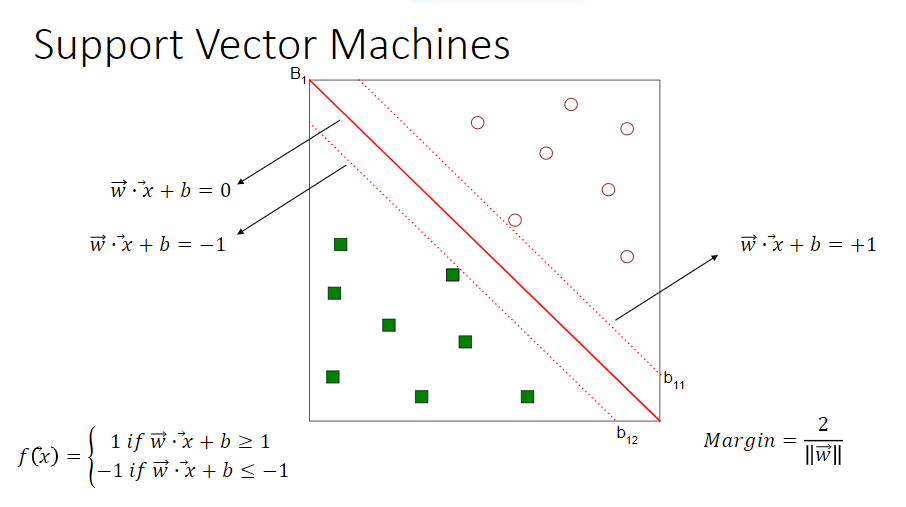
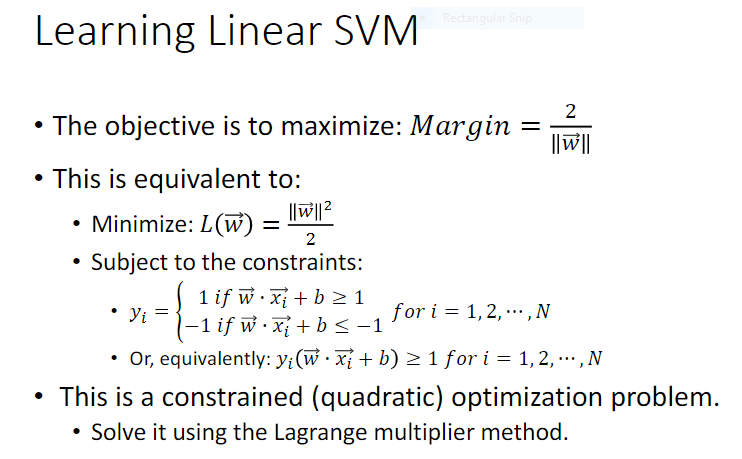
**Ensemble:**

* Same as random forests, except you can use any classification algorithm (can add weights)

**Characteristics of ANN:**

* Multilayer ANN are universal approximators.
* They can suffer from overfitting if the network is too large.
* Gradient descent may converge to local minimum.
* Model building can be very time consuming, but testing can be very fast.
* Can handle redundant attributes because weights are automatically learned.
* Sensitive to noise in training data.
* Difficult to handle missing attributes.

**SVM:**

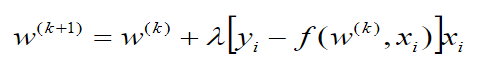
****

* If the data can’t be classified linearly, transform to higher dimension.
* Objective is to minimize the slack variable too.

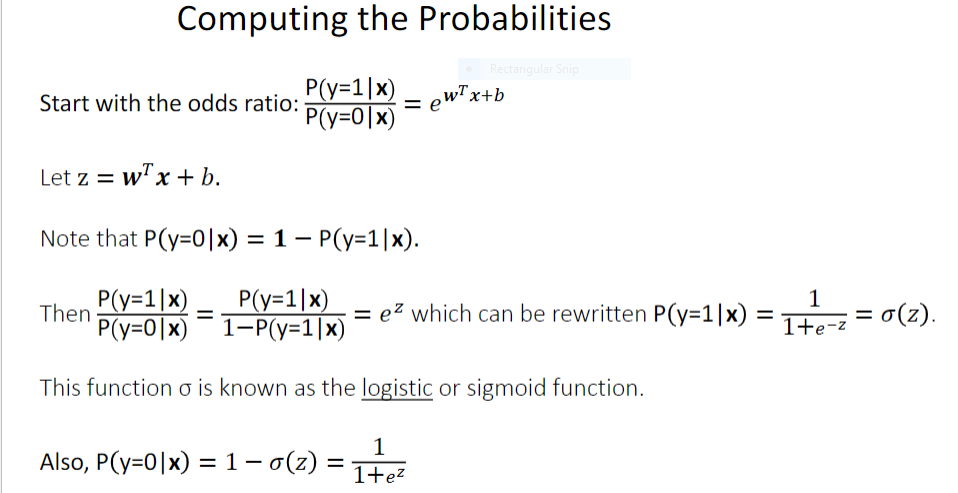
Characteristics of SVM:

* Since the learning problem is formulated as a convex optimization problem, efficient algorithms are available to find the global minima of the objective function (many of the other methods use greedy approaches and find locally optimal solutions)
* Overfitting is addressed by maximizing the margin of the decision boundary, but the user still needs to provide the type of kernel function and cost function.
* Difficult to handle missing values
* Robust to noise
* High computational complexity for building the model

**ANN:**

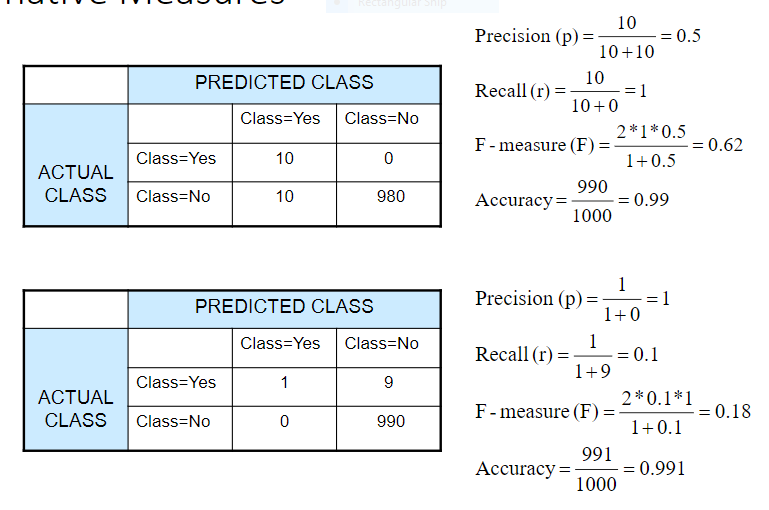
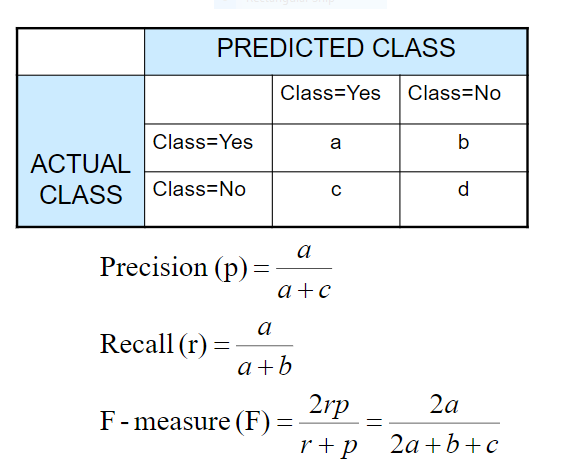
****

**Logistic Regression:**

****

Coverage = No. of records that satisfy the predictor / Total number of records

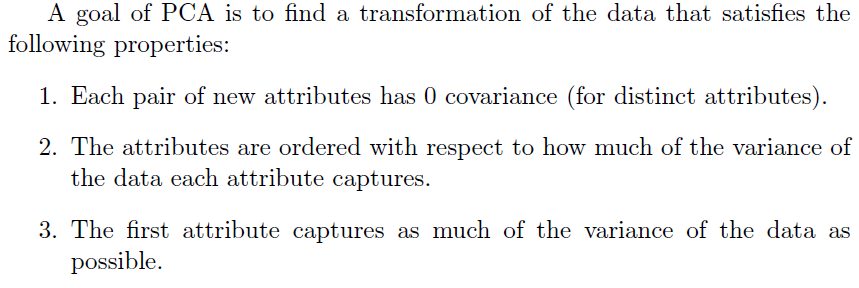
Accuracy = No. of records that satisfy the consequent / No. of satisfied predictors

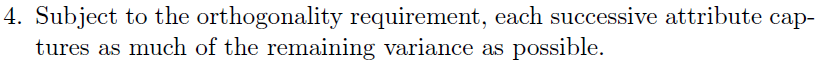


**PCA:**

* Finds a projection of the high dimensional data along an axis with the highest variance. (new data points are uncorrelated)
* Centers the data by subtracting mean from all data points to center the data.
* Tends to identify strong patterns in data. Can be used as a pattern finding technique.
* Deals with curse of dimensionality, removes redundant/irrelevant features/noise, more visualizable.
* 2D representation -> First dimension - max variability as possible, second dim – remaining variability.
* Pick number of dimensions based on how much variance they capture.

Goal: Covariance Matrix





**SVD:** Does not center data. SVD is used to do PCA.

**MDS:** Preserves distances between points in high dimensional space and low dimensional space. (knn on a place)

**ISOMAP:** Extension of MDS. Deals with non-linear data.