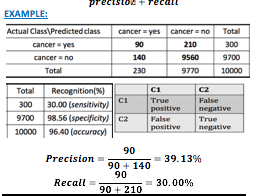
**Classification**: Goal is to organize and categorize data in distinct classes. We know the class labels and the number of classes - two step process: Model construction and model usage(testing the model against other data and using in prediction) Phases of building a classifier: 1. divide data into training and test data. 2. Induce a classifier. 3. Test 4. Use to predict new values

->Preparing Data for Classification: **1.Data Preparation**: Transformation(Discretization, normalization), Cleaning(Smoothing reduce noise), Relevance Analysis (Feature selection-eliminate irrelevant attributes) **2.Data Evaluation:** Evaluating classification methods -Accuracy(classifier/ predictor)-Speed(training time, classification/prediction time)-Robustness(handling noise/missing vals)-Scalability(efficiency) -Interpretability(understanding & insight provided by model) Accuracy rate: percentage of test set samples that are correctly classified.

->Model Evaluation & Selection: Is used to evaluate quality of model

-->Accuracy measures: **1.Accuracy:** % of tuples correctly classified by model M **2.Error rate** = 1-(acc(M))

3.sensitivity/recall/TPR = true pos/(true pos+false pos) 4)Specificity=true neg/(true neg+false pos) 5)F-measure=(2\*prec\*recall)/(prec+recall) [Inverse relationship precision & recall] 

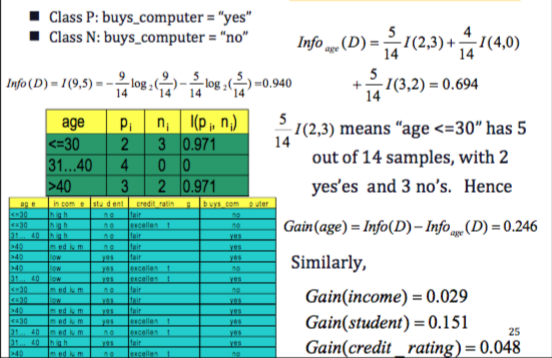
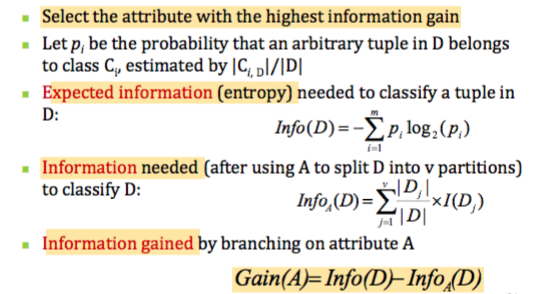
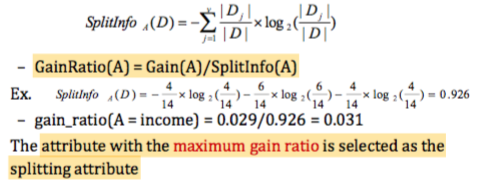
SAMPLING class imbalance: Used to balance frequencies of instances of all classes in dataset: 1)OverSampling- Instances of minority classes introduced into training set 2)UnderSampling - Instances of majority classes removed (causes info loss)

Accuracy of Classifier: **Holdout method:** data randomly partitioned into two independent. (Random sampling - variation of holdout) (accuracy of holdout repeated k times) **Cross-validation:** k-fold (10-fold). Randomly partition the data into k mutually exclusive subsets of approx equal size. Repeat multiple times with different hold-out or test sets. Evaluate significance of results: use t-test for pairwise comparisons.

Supervised Learning :Focuses on classification: choosing among subgroups to best describe a new instance of data, and prediction, which involves estimating an unknown parameter. Split data in training and testing so no false results, which avoids overfitting. No one best solution. best solution would involve fitting a model that matches the underlying model that generated the data. Unfortunately, we typically have no idea what that the underlying model is. An empirical solution is to try a number of algorithms, making sure to keep aside test data on which to evaluate performance, but that can be time consuming. accepting the limitations of each class of algorithms, can use a process called blending, merging the outputs of multiple different algorithms.

-> Decision Trees: Internal nodes denote attribute-value test, branches denotes the different outcomes of the test. Leaf node represents class label. **Advantages**:easy to interpret, can handle numerical and categorical. **Disadvantages**:- Which split criterion(goodness functions) -Branching scheme: determining the tree branch to which a sample belongs - Stopping decision when to stop the further splitting of a node- Labeling rule: which class to label a node as. Overfitting: Pruning-technique that reduces the size of the decision trees by removing sections of the tree that provide little power to classify instances. reduces complexity of final classifier, & improves predictive accuracy by reduction of overfitting.1)Pre Pruning:during construction - best choice 2)Post:after-top/down

-->Top-down induction: **1.Tree construction:** select an attribute to partition. Recursively select attributes and do partition. **2.Tree pruning:** Tree may be reclassified. Remove tree branches that may reflect noise. Improve classification accuracy.

-->Information gain: assume all attributes are categorical. Need to determine the best split point for A. Sort A. midpoint between each pair of adjacent values is considered as a possible split point. Point with *minimum expected information requirement* for A is selected as the split-point for A. It is biased towards attributes with a large number of values. To overcome that problem C4.5 was created, it uses gain ratio. 

-->Gini index (CART): assume all attributes are continuous. Several possible split points for each attribute.

-->Random Forest:Random forests are constructed from decision trees. A notable difference between each tree is that each only has access to a subset of training examples, a concept known as bagging. Then each tree is allowed to cast an independent vote on a final classification and serves as a means of regularization. More robust than adaBoost. Faster than bagging or boosting. **Advantages:** Avoids overfitting, can be used to identify the most important features from the training dataset. **Disadvantages:** Model Size(Large memory allocation)

-> Support Vector Machines (SVM): Searches for Maximum Marginal Hyperplane. Finds the hyperplane using support vectors. Designed for ‘Binary Classification Problems’. If data is not linearly separable, transform original input data into a higher dimensional space. SVM allows to distinguish classes for all new vectors. It is effective on high dimensional data. Complexity of trained classifier is characterized by the # of support vectors rather than the dimensionality of the data. **Advantages:** Avoids overfitting, **Disadvantages:** Kernel Parameters and choice of kernel

-->Support Vectors: Any training tuples that fall on the hyperplanes margin. The support vectors constrain the width of the margin.

->Ensemble Modeling: Machine Learning CONCEPT in which we train multiple models using the same learning algorithm. The ‘ensembles’ take part in a bigger group of methods, called multi-classifiers. Ensemble helps to minimize noise, bias and variance.

-->Multi-classifiers:Large set (hundreds-thousands) of learners with common objective are fused together to solve

--->Bagging: averaging the prediction over a collection of classifiers. Majority vote. Any element has the same probability to appear in a new data set. Training stage is parallel for Bagging. Bagging takes simple average. More robust. Improved accuracy in prediction.

--->Boosting: weighted vote with a collection of classifiers. Weight assigned based on the previous diagnosis accuracy. Observations are weighted and therefore some of them will take part in the new sets more often. Boosting builds the new learner in a sequential way. Boosting takes weighted average. Better accuracy but risks overfitting.

---->AdaBoost: AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers

--->Stacking: Combining a set of heterogeneous classifiers. Several opinion given to final classifier to make recommendation. Multi-view.

->Lazy vs. Eager Learning: **Lazy Learning**: Simply stores training data (minor processing) and waits until it is given a test tuple. Less time in training but more time in predicting. **Eager Learning**: Given a set of training tuples, constructs a classification model before receiving data. Must commit to a single hypothesis that covers the entire instance space. -->Eager Learning: Munges the training data as soon as it receives it (Supervised Classification). Given a set of training tuples, constructs a classification model before receiving new (e.g, test) data to classify -->Lazy: Simply Stores the training data without doing any further munging on it. K-NN most common.

--->K-Nearest Neighbor Algorithm: Assigns class based on the values of the most similar training examples.Odd data samples are used to avoid ties. Returns the mean values of the k nearest neighbors. Uses euclidean distance. **Advantages:** Robust to noisy data, **Disadvantages:** Curse of dimensionality, distance between neighbors could be dominated by irrelevant attributes

**All of these algorithms have their pros and cons, according to the so‐called No Free Lunch theorem. An advantage of decision trees is that it is easy to understand. Also, it makes the decision process very easy to understand, and give users the ability to branch where needed. That is, decision makers can clearly see the impact of following specific paths in the tree. However, if data are noisy, the results may be less accurate and this may also lead to a bushy tree. If the tree is too complex, then it is difficult to follow the decision flow and make decisions where to branch.**

**SVMs are generally good at learning in domains where the data are more difficult to classify and provide accurate results. Also, it is able to even find models for non‐linear data, using the “kernel trick”. However, the results of SVMs are not so easy to interpret and they may also take quite some time to train against larger datasets. Further, it is sometimes a challenge to find the hyperplane in noisier data where classes overlap.**

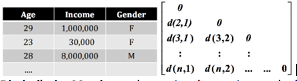
**The k‐NN algorithm is very flexible since it learns as instances arrive. It is also flexible to the distance measure you choose. In large datasets, finding the nearest neighbors may involve a very large search problem. The algorithm is very sensitive both to the choice of k and the distance function. A concern is that one cannot interpret the results**.

**Chapter 7: Cluster Analysis (unsupervised learning)**

Cluster: collection of data objects similar to one another in the same group, and dissimilar to objects in other groups

->Cluster analysis: grouping similar data objects into clusters. Unsupervised learning because no predefined classes -intra-cluster distances are minimized -inter-cluster distances are maximized

->Quality of clustering depends on: - similarity measure used by method - its implementation - ability to discover hidden patterns, - high intra-class similarity (cohesive within clusters), low inter-class similarity (distinctive between clusters)

Measure of Similarity: Numerical measure of how alike two data objects are - Value is higher when objects are more alike - Often falls in the range [0,1], [0.1,0.9] or [-1,1].

->Measure of Dissimilarity (Proximity in clustering): Numerical measure of how different two data objects are - value higher when objects more alike.

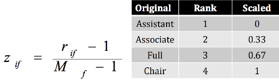
dissimilarity often 0 - upper limit varies

->Proximity refers to a similarity or dissimilarity

Data Matrix: n data points with p dimension - two modes

Dissimilarity Matrix: - n data points but register only the distance - one mode

Proximity measure for nominal attributes: Create a new binary attribute for each of the M nominal states, ie {orange, yellow, blue} convert to {y/n,y/n,y/n}

Ordinal value (with rank): treat like interval scaled - map the range of each variable onto [0,1] with

-->k-means for clustering: **1.Partition** objects into k nonempty subsets. **2.compute** seed points as the centroids of the clusters of the current partitioning (the centroid is the center (**ie mean point** of the clusters). **3.Assign** each object to the cluster with the nearest seed point. **4.Back to step 2** until assignment doesn’t change. **Advantages:** efficient- **O(tkn)**, n=#objects, k=#clusters, t=#iterations. Normally k,t<<n. Often terminates at a local optimal. **Disadvantages:** -applicable only to objects in n-dimensional space.-must specify k in advance. -sensitive to noisy data and outliers. -not suitable to discover clusters with non-convex shapes. -->Assessing cluster tendency: test spatial randomness by statistic test: **Hopkins static**. -given dataset D, with sample random variable o, determine how far o is from being uniformly distributed in the data space. (No equation) if D uniformly distributed, H is close to 0.5, if highly skewed: H close to 0. -->Determine number of clusters: **1.empirical:** #clusters=sqrt(n/2) for dataset of n points. **2. Elbow:** use turning point in curve of sum within cluster variance w.r.t #clusters. **3.Cross validation:** divide dataset into m parts, use m-1 parts to obtain clustering model, use remaining part to test quality of clustering, for any k>0 repeat it m times, compare quality measure wrt different k’s and find #clusters that fits data best. **→ Solution to initial centroid problem:** Multiple runs. Select more than k initial centroids then select among these initial centroids. Post-processesign. Generate a larger number of clusters then merge

→ Evaluation clusters k-means: **SumofSquaredError(SSE):** for every point, error is the distance to nearest cluster, to get sse: square these errors and sum them, given two sets of clusters we prefer the one with smallest error, **one way to reduce sse** is to increase k(good clustering with smaller K can have lower SSE tham poor clustering. with higher K). -->measuring clustering quality: **1.extrinsic:** supervised, ground truth available, -compare clustering against ground truth using certain clustering measure. **quality measure** good if follows 4 criteria: **- 1.1 Cluster homogeneity:** thePurer the better**, -1.2 Cluster completeness:** Assigning ground truth to same cluster **-1.3 Rag bag**:put heterogeneous object into pure object, should be penalized more than putting in other **-1.4 Small cluster preservation:**Splitting clusters into small piece more harmful than splitting large **2. Intrinsic:** unsupervised, ground truth unavailable. -evaluate the goodness of a clustering by how well clusters are separated and how compact they are

--->Density-based: clustering based on density. **Two parameters:** **eps**-maximum radius of neighbourhood, **minpts**-minimum number of points in the eps-neighbourhood of that point. **-->DBSCAN**: relies on density-based notion of cluster -a cluster is defined as a maximal set of density connected points -discovers clusters of arbitrary shape in databases with noise. **DBSCAN Algorithm:** arbitrary select point p, retrieve all points density-reachable from p w.r.t EPS and MINPTS, if p is a core point: cluster is formed, if p is a border point: no points are density reachable from p and DBSCAN visits the next point of the database, continue the process until all points have been processed.

**Fuzzy Set** and **Fuzzy clusters**: some applications may need **fuzzy** or **soft clusters assignment** such as “e-game could belong to both entertainment and software”

**Probabilistic clustering:** cluster analysis to find hidden categories, hidden category is a distribution over the data space which can be mathematically represented using a probability density function.

**Expectation-Maximum (EM) algorithm:** framework to approach maximum likelihood of parameters in statistical models: - **E-steps** assigns objects to clusters according to the current fuzzy clustering or **parameters of probabilistic clusters**. - **M-steps** find the new clustering or parameters that maximize the sum of squared error (SSE) or the expected likelihood.

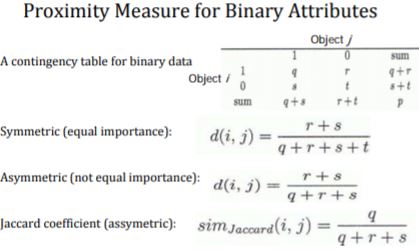
**Clustering high-dimensional data:** applications such as text documents, DNA micro-array data - **method** - **subspace-clustering**: search for clusters existing in subspaces of the given high dimensional data space, eg CLIQUE **- dimensionality reduction approaches**

**Major challenges:** many irrelevant dimension may mask clusters - distance measure becomes meaningless du to equi-distance - clusters may exist only in one subspaces - traditional distance measure could be dominated by noises in many dimensions

**Curse of dimensionality:** data in only one dimension is packed - adding a dimension stretches out the points across that dimension, making them further apart - high dimensional data is extremely sparse

**Subspace clustering:** clusters may exists only in some subspaces **method:** bi-clustering, cluster both objects and attributes at the same time. **4 requirements:** only small set of objects participate in a cluster - cluster only involves small set of attributes - object may be involved in multiple or no clusters at all

**Issues:** clusters may not be exclusive - no globally defined similarity/distance measure

--->EM vs.K-means The main 

difference between k‐means and EM is that the clusters in EM may overlap. In class, we discussed using the Euclidean distance and fuzzy clustering within EM.The k‐means algorithm works best for data that form spherical, convex clusters with regular shapes. It is not great with noisy data or data containing outliers, since it will attempt to fit the noise or outlier into the model.

EM is better if we want the possibility that the clusters may overlap, and/or the data do not fix the convex‐shaped model.

The k‐means algorithm works best for data that form spherical, convex clusters with regular shapes. It is not great with noisy data or data containing outliers, since it will attempt to fit the noise or outlier into the model. EM is better if we want the possibility that the clusters may overlap, and/or the data do not fix the convex‐shaped 