Correlation is more on line of regression

Covariance is strength of direction

**Proportionate Sampling**: Using pandas groupby, separate the students into groups based on their grade i.e A, B, C, and random sample from each group based on population proportion. The total sample size is 60%(0.6) of the population

**Classification definition**

Given a collection of records (*training set* )

–Each record contains a set of *attributes*, one of the attributes is the *class*.

lFind a *model* for class attribute as a function of the values of other attributes.

Goal: previously unseen records should be assigned a class as accurately as possible.

–A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

**Clustering definition**

lGiven a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that

–Data points in one cluster are more similar to one another.

–Data points in separate clusters are less similar to one another.

**Sequential Pattern Discovery: Definition**

lGiven is a set of *objects*, with each object associated with its own *timeline of events*, find rules that predict strong sequential dependencies among different events.

lRules are formed by first disovering patterns. Event occurrences in the patterns are governed by timing constraints.

**Association definition**

lGiven a set of records each of which contain some number of items from a given collection;

–Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

**Clustering vs classification**

Clustering

lGoal is to identify similar groups of objects

lGroups (clusters, new classes) are discovered

lInput: Flat File dataset

lUnsupervised (class label has to be learned)

lImportant: Similarity assessment which derives a “*distance function*” is critical, because clusters are discovered based on distances/density.

Classification

lPre-defined classes

lInput: Flat File dataset consist of attributes and a specific class attributge

lSupervised (class label is given)

lGoal is to predict classes from the object properties/attribute values

lClassifiers are learnt from sets of classified examples

lImportant: classifiers need to have a high accuracy

**Graph mining**

lGoal: Find interesting and/or frequent patterns in (large) graphs; the purpose is to extract patters (sub-graphs) of interest from graphs.

**Neural net Graph**

lHow do they do that? The create graph embeddings which map the nodes/edges/the whole graph in fixed length vectors of numbers and NNs the learn on those vectors in flat file format.

**Data curation**

is a term used to indicate management activities related to organization and integration

of data collected from various sources, annotation of the data, and publication and presentation of the data such that the value of the data is maintained over time, and the

data remains available for reuse and preservation.

**Missing values**

lReasons for missing values

–Information is not collected

(e.g., people decline to give their age and weight)

–Attributes may not be applicable to all cases

(e.g., annual income is not applicable to children)

–

lHandling missing values

–Eliminate Data Objects

–Estimate Missing Values

–Ignore the Missing Value During Analysis

–Replace with all possible values (weighted by their probabilities)

–Learn a model that predicts the missing values

**Decision Boundary of Decision Trees**

• Border line between two neighboring regions of different classes is known as decision boundary

• Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

**Histograms**

Modal

Skewed

Peaks

No GAPS

Outliers

A histogram with peaks pressed up against the graph “walls” indicates a loss of information, which is nearly always bad.

So check both the right and left ends of the histogram. Look for any clipping – highlight clipping along the right side, and shadow clipping along the left side.

–What is the type of the attribute? Positive real numbers

–What is the mean value; what is the mode?

–Is the a lot of spread or not (compute the standard deviation)? Not much

–Is the distribution unimodal (one hill or no hill)) or multi-modal (multiple hills)? One hill or two hills, depending on how you interpret the data. The second hill is not very well separated; therefore I would say unimodal.

–Is the distribution skewed (e.g. compare mean with median)? Only very mildly skewed

–Are there any outliers? Yes values above 45…?

–Are there any duplicate values?

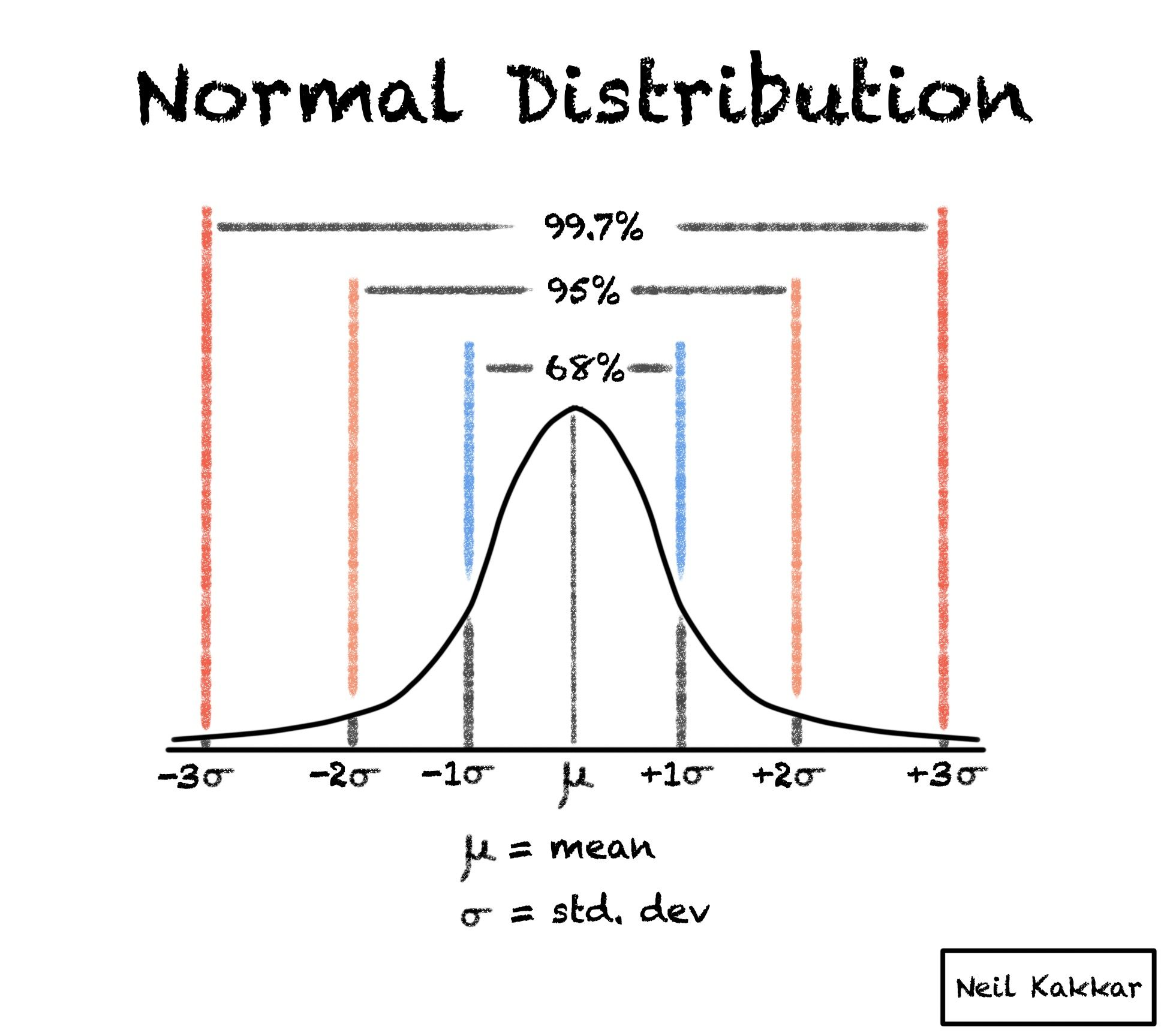
–Are there any gaps in the attribute value distribution? Yes two gaps: 1)… 2)…

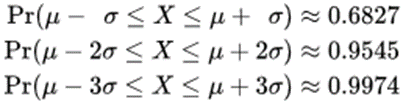
–Characterize the shape of the density function! Bell Curve

–

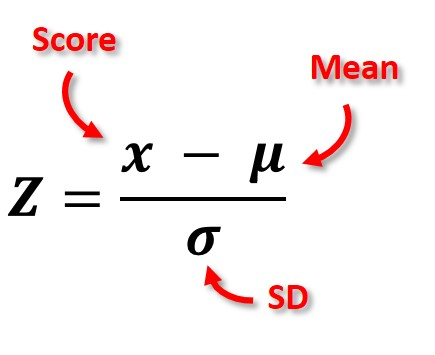
–

**99.7 rule**





**Z-score**



lHaving the value of 1 in [0,1] normalization indicates the maximum value, and a value of 0 indicates the minimum value and a value of 0.5 indicates … What??

lA z-score of -2 indicates a value 2 standard deviations below the mean value, a value of 0 indicates the mean value of the attribute and a z-score of +1 indicates a value one standard deviation above the mean value.

lAs z-scores allow for statistical interpretation they are usually the preferred choice, but some scientists prefer [0,1] normalization, as they do not like to work with negative values in the dataset.

lDatasets are frequently standardized before applying a data mining technique to the dataset! Why??

Box Plot

Questions to ask:

1.Compare the two medians: are the two medians: more or less the same: at most 1/10 box\_size apart, somewhat similar: 0.1 to 0.25 box\_size apart, different: 0.25 to 1 box\_size apart, very different: more than 1 box\_size apart

2.Compare the IQRs of the two box plots by interpreting them as intervals, computing their overlap (overlap:=length of the intersection of the intervals they cover/length of the union of the intervals they cover): 90-100%: IQRs almost the same; 75-90%: quite similar, 25-75: somewhat overlapping, 25-10%: dissimilar, 10% or less: very dissimilar.

3.Compare the intervals defined by the two whiskers (WI) of the two boxplots by computing their WI overlap, and draw conclusions as follows: Range of attribute is: 90-100%: almost the same; 75-90%: quite similar, 25-75%: has medium overlap, 25-10%: dissimilar, 10% or less: very dissimilar.

4.Assess and compare skewness by summarizing the position of the median in the box

5.Mention and assess agreement/disagreement with respect to outliers that occur in both boxplots

6.Combine the agreement/disagreement in the 2 boxplots you observed by transforming the answers to questions 1-5 into a final summary.

Heuristic: We use the IQR as a proxy for standard deviation when comparing the respective distributions, and set box\_size to (IQR1+IQR2)/2!

Supervised scatterplot

lInterpreting Scatter Plots and Similar Display

–Characterize the distribution of each class in the attribute space; is it unimodal or mult-imodal? Unimodal each.

–Characterize the overall distribution (including all examples); do you observe any correlation or other characteristics? quite strong positive correlation between the two attributes.

–Analyze the separation of a single class from all the other classes. Analyze the separation between pairs of classes. Blue is clearly separated from the two other; red and green only slightly overlap;

–If classes overlap characterize the extend to which they overlap.

–If decision boundaries between classes can be inferred characterize those decision boundaries. Test using just sepal length will mostly do a good job.

–Assess the difficulty of the classification based on your findings of looking at a set of scatter plots. Easy

**Decision Tree**

•Used for classifying data by partitioning attribute space

•Tries to find axis-parallel decision boundaries for specified optimality criteria

•Leaf nodes contain class labels, representing classification decisions

•Keeps splitting nodes based on split criterion, such as GINI index, information gain or entropy

•Pruning necessary to avoid overfitting

**Hunt’s algorithm**

oLet Dt be the set of training records that reach a node t

oGeneral Procedure:

–If Dt contains records that belong the same class yt, then t is a leaf node labeled as yt

–If Dt is an empty set, then t is a leaf node labeled by the default class, yd

–If Dt contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

**Greedy strategy**

–Split the records based on an attribute test that optimizes certain criterion.

oCreates the tree top down starting from the root, and splits the records based on an attribute test that optimizes certain criterion.

oIssues

–Determine how to split the records

u How to specify the attribute test condition?

u How to determine the best split?

–Determine when to stop splitting

oMakes locally optimal choices at each stage

oFast and therefore attractive to solve NP-hard and other problems with high complexity. Later decisions are made in the context of decision selected early dramatically reducing the size of the search space.

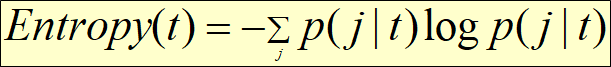
oThey do not backtrack: if they make a bad decision (based on local criteria), they never revise the decision.

oThey are not guaranteed to find the optimal solution(s), and sometimes can get deceived and find really bad solutions.

oIn spite of what is said above, a lot successful and popular algorithms in Computer Science are greedy algorithms.

**Entropy**

oEntropy at a given node t:



(NOTE: *p( j | t)* is the relative frequency of class j at node t).

–Measures homogeneity of a node.

uMaximum (log nc) when records are equally distributed among all classes implying least information

uMinimum (0.0) when all records belong to one class, implying most information

–Entropy based computations are similar to the GINI index computations

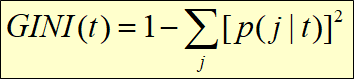
**Gain**

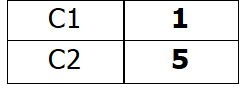
**Decision tree**

Impurity/non-homo = 5/5

Purity/homo = 10/0

**GINI Index**





**P(C1) = 1/6 P(C2) = 5/6**

**Gini = 1 – (1/6)2 – (5/6)2 = 0.278**

**When a node only has one option to go into its Gini index is 0**

**Because it’s a pure node**

oUse Binary Decisions based on one value

oSeveral Choices for the splitting value

–Number of possible splitting values

= Number of distinct values

oEach splitting value has a count matrix associated with it

–Class counts in each of the partitions, A < v and A ³ v

oSimple method to choose best v

–For each v, scan the database to gather count matrix and compute its Gini index

–Computationally Inefficient! Repetition of work.

**Entropy and gain**

Assume we have m classes in our classification problem. A test S subdivides the examples D= (p1,…,pm) into n subsets D1 =(p11,…,p1m) ,…,Dn =(p11,…,p1m). The qualify of S is evaluated using Gain(D,S) (ID3) or GainRatio(D,S) (C5.0):

Let H(D=(p1,…,pm))= Si=1 (pi \*log2(1/pi)) (called the entropy function)

Gain(D,S)= H(D) - Si=1 (|Di|/|D|)\*H(Di)

Gain\_Ratio(D,S)= Gain(D,S) / H(|D1|/|D|,…, |Dn|/|D|)

Remarks:

|D| denotes the number of elements in set D.

D=(p1,…,pm) implies that p1+…+ pm =1 and indicates that of the |D| examples p1\*|D| examples belong to the first class, p2\*|D| examples belong to the second class,…, and pm\*|D| belong the m-th (last) class.

H(0,1)=H(1,0)=0; H(1/2,1/2)=1, H(1/4,1/4,1/4,1/4)=2, H(1/p,…,1/p)=log2(p).

C5.0 selects the test S with the highest value for Gain\_Ratio(D,S), whereas ID3 picks the test S for the examples in set D with the highest value for Gain (D,S).

For example, a dataset with 30 Positive and 70 Negative samples has its Entropy:

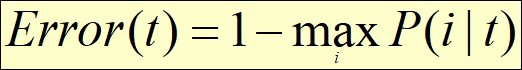


**oClassification error at a node t :**

oMeasures misclassification error made by a node.

uMaximum (1 - 1/nc) when records are equally distributed among all classes, implying least interesting information

uMinimum (0.0) when all records belong to one class, implying most interesting information



**Tree Induction**

oStop expanding a node when all the records belong to the same class

oStop expanding a node when all the records have similar attribute values

oEarly termination (used by many decision tree tools to avoid overfitting); examples include:

–Terminate the number of examples left is less than x (e.g. x=4)

–Terminate if the depth of the tree has reached a limit (e.g. limit=6)

–Terminate if the purity of the examples associated with a node is x% or more (e.g. x=90)

o

**Underfitting**: when model is too simple, both training and test errors are large

**Overfitting**: when model is too complex and test errors are large although training errors are small.

**Advantages of Decision tree**

oInexpensive to construct

oExtremely fast at classifying unknown records

oEasy to interpret for small-sized trees; strategy understandable to domain expert

oOkay for noisy data

oCan handle both continuous and symbolic attributes

oUseful for exploratory data analysis

oAccuracy is comparable to other classification techniques for many simple data sets

oDecent average performance over many datasets

oCan handle multi-modal class distributions

o Kind of a standard—if you want to show that your “new” classification technique really “improves the world” à compare its performance against decision trees (e.g. C 5.0) using 10-fold cross-validation

oDoes not need distance functions; only the order of attribute values is important for classification: 0.1,0.2,0.3 and 0.331,0.332, and 0.333 is the same for a decision tree learner.

Disadvantages

oRelies on rectangular approximation that might not be good for some dataset

oSelecting good learning algorithm parameters (e.g. degree of pruning) is non-trivial; however, some of the competing methods have worth parameter selection problems

oEnsemble techniques, neural networks, support vector machines, and sometimes kNN, might obtain higher accuracies for a specific dataset.

–

**Regression Tree**

oRegression trees work as decision trees except

–Leafs carry numbers which predict the output variable instead of class label

–If a leaf is created the average value of the dependent variables of the examples associated with the leaf node is used as the prediction of the leaf node.

–Instead of entropy/purity the squared prediction error serves as the performance measure.

–Tests that reduce the squared prediction error the most are selected as node test; test conditions have the form y>threshold where y is an independent variable of the prediction problem.

–Instead of using the majority class, leaf labels are generated by averaging over the values of the dependent variable of the examples that are associated with the particular class node

oLike decision trees, regression tree employ rectangular tessellations with a fixed number being associated with each rectangle; the number is the output for the inputs which lie inside the rectangle.

oIn contrast to ordinary regression that performs curve fitting, regression tries use averaging with respect to the output variable when making predictions; the output for inputs of a particular patch is constant.

**–**

**–**

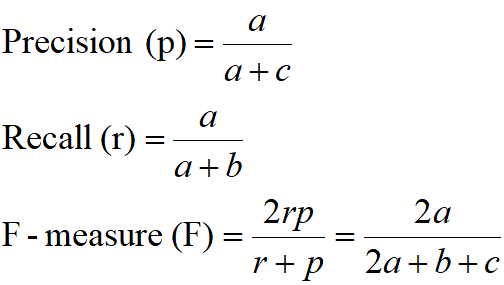
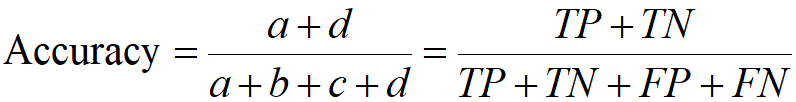
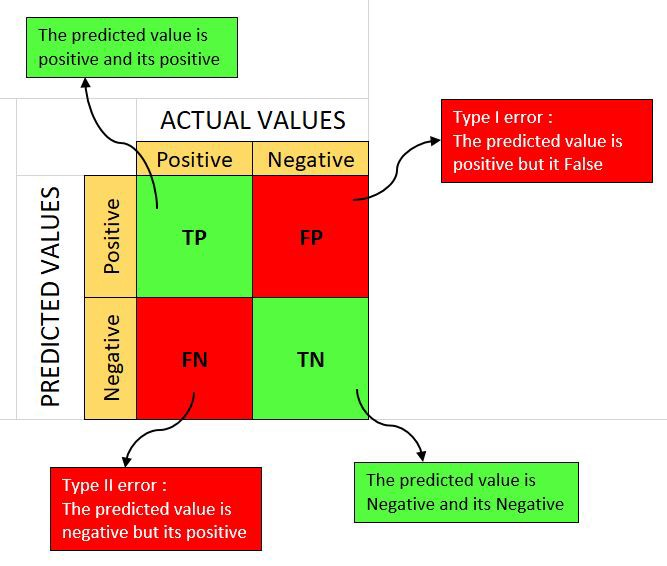
**N-fold**

o10-fold cross validation is the most popular technique to evaluate classifiers based on a performance metric (e.g. accuracy)

oLeave one out and stratified cross validation also has some popularity

oCross validation is usually performed class stratified (frequencies of examples of a particular class are approximately the same in each fold).

oExample should be assigned to folds randomly (if not à *cheating*!)

oAccuracy:= % of testing examples classified correctly

lPre-Pruning (Early Stopping Rule)

–Stop the algorithm before it becomes a fully-grown tree

–Typical stopping conditions for a node:

u Stop if all instances belong to the same class

u Stop if all the attribute values are the same

–More restrictive conditions:

u Stop if number of instances is less than some user-specified threshold

u Stop if class distribution of instances are independent of the available features (e.g., using c 2 test)

u Stop if expanding the current node does not improve impurity

measures (e.g., Gini or information gain).

lPost-pruning

–Grow decision tree to its entirety

–Trim the nodes of the decision tree in a bottom-up fashion

–If generalization error improves after trimming, replace sub-tree by a leaf node.

–Class label of leaf node is determined from majority class of instances in the sub-tree

**Overfitting**

lOverfitting results in decision trees/models that are more complex than necessary: after learning knowledge they “tend to learn noise”

lMore complex models tend to have more complicated decision boundaries and tend to be more sensitive to noise, missing examples,…

lWhen learning complex models, large representative training sets are needed.

lFor small datasets, simple models are often a “good choice”.

lIn summary, the two approaches to fight overfitting are:

–Reduce model complexity

–Increase the size of the training set

**Nearest Neighbor**

lRequires three things

–The set of stored records

–Distance Metric to compute distance between records

–The value of *k*, the number of nearest neighbors to retrieve

–

lTo classify an unknown record:

–Compute distance to other training records

–Identify *k* nearest neighbors

–Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

**k-NN**

§k-NN classifiers are lazy learners

–Unlike eager learners such as decision tree induction and rule-based systems, it does not build models explicitly

–Classifying unknown records is relatively expensive

§k-NN classifiers rely on a distance function; the quality of the distance function is critical for the performance of a K-NN classifier.

§k-NN classifiers obtain high accuracies and are quite popular in some fields, such as text data mining and in information retrieval, in general.

**Margin**

§Find hyperplane maximizes the margin => B1 is better than B2

**Vector machines**

§Support vector machines learn hyperplanes that separate two classes maximizing the *margin between them* (*the empty space between the instances of the two classes)*.

§Support vector machines introduce slack variables, in the case that classes are not linear separable and trying to maximize margins while keeping the training error low.

§The most popular versions of SVMs use non-linear kernel functions to map the attribute space into a higher dimensional space to facilitate finding “good” linear decision boundaries in the modified space.

§Support vector machines find “margin optimal” hyperplanes by solving a convex quadratic optimization problem. The complexity of this optimization problem is high; however, as computer got faster using SVMs on large datasets is no longer a major challenge.

§In general, support vector machines accomplish quite high accuracies, if compared to other techniques.

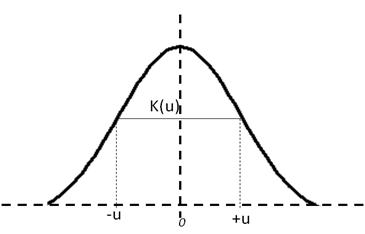
§

C.R.A.P stands for contrast, repetition, alignment and proximity

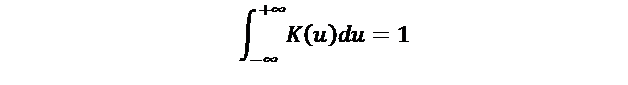
**Kernel**

Kernel is simply a function which satisfies following three properties as mentioned below. Kernel functions are used to estimate density of random variables and as weighing function in non-parametric regression. This function is also used in machine learning as kernel method to perform classification and clustering.

1. The first property of a kernel function is that **it must be symmetrical**. This means the values of kernel function is same for both +u and –u as shown in the plot below. This can be mathematically expressed as K (-u) = K (+u). The symmetric property of kernel function enables the maximum value of the function (*max(K(u)*) to lie in the middle of the curve.



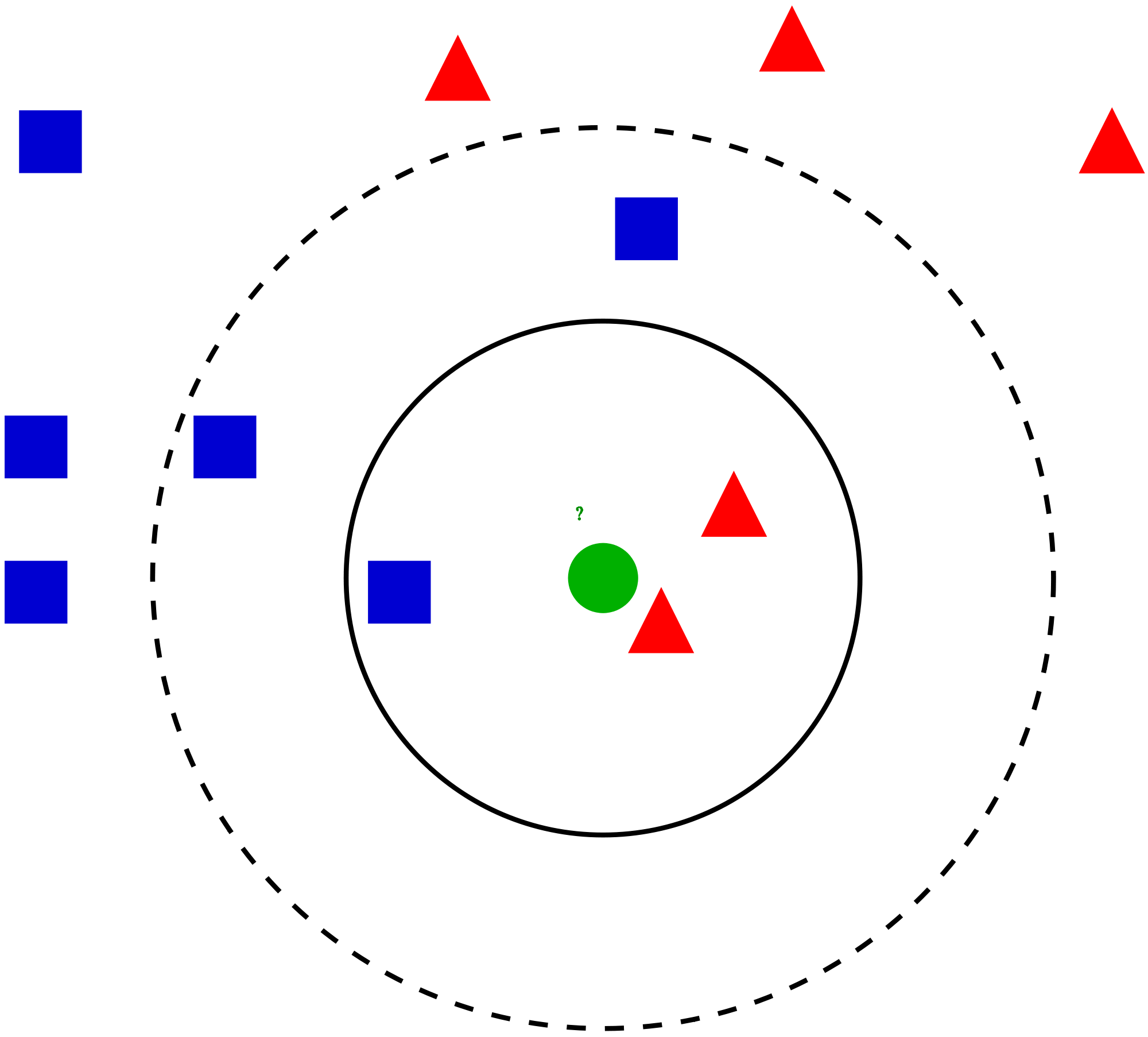
2. The **area under the curve of the function must be equal to one**. Mathematically, this property is expressed as



Gaussian density function is used as a kernel function because the area under Gaussian density curve is one and it is symmetrical too.

3. The value of kernel function, which is the density, **can not be negative**, K(u) ≥ 0 for all −∞ < u < ∞.

**k-NN**



**Computing error**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0