Matters of Discussion

Classifications - Classification methods

Decision Tree,

Naïve Bayes,

K-Nearest Neighbors

Classification And Regression Trees –Logistic Regression Models. [To be discussed Later]

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Prediction Problems: Classification

- Classification
  - predicts categorical class labels
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Typical applications
  - Credit/loan approval:
  - Medical diagnosis: if a tumor is cancerous or benign
  - Fraud detection: if a transaction is fraudulent
  - Web page categorization: which category it is

Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction is training set
  - The model is represented as classification rules, decision trees, or mathematical formulae

Cont..

- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set
    - If the accuracy is acceptable, use the model to classify new data.

Process (1): Model Construction

Training Data

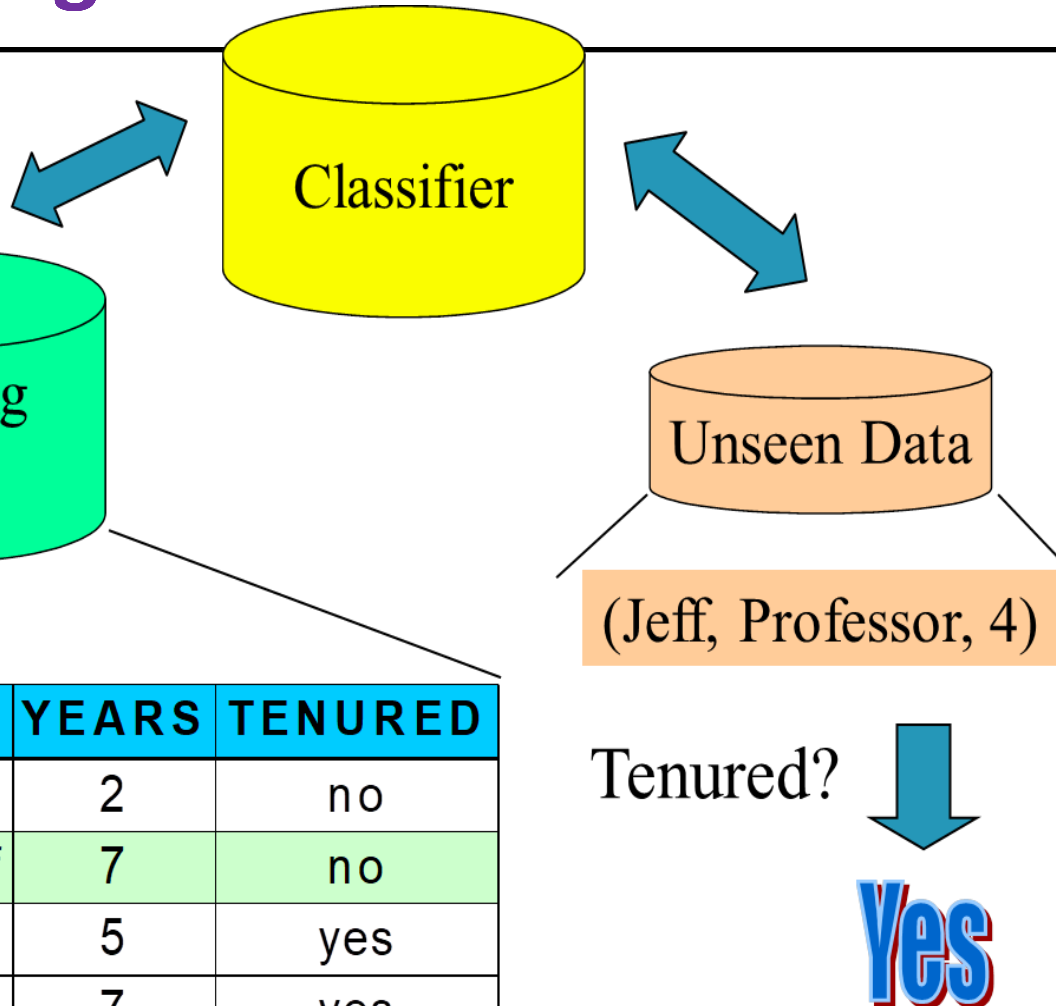
Classification Algorithms

Classifier (Model)

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

Process (2): Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Performance measures

❖The performance of the developed model can be evaluated using **Confusion Matrix**

		Predicted Class	
		Class = **Positive**	Class =**Negative**
Actual Class	**Class = Positive**	True Positive (TP)	False Negative (FN)
	Class = Negative	False Positive (FP)	True Negative (TN)

Performance measures

- Performance metrics

$$Sensitivity\ (\%) = \frac{TP}{TP+FN}\ X\ 100$$

$$Specificity\ (\%) = \frac{TN}{TN+FP}\ X\ 100$$

$$Precision\ (\%) = \frac{TP}{TP+FP}\ X100$$

$$Recall\ (\%) = Sensistivity\ (\%) = \frac{TP}{TP+FN}\ X\ 100$$

$$Accuracy\ (\%) = \frac{TP+TN}{TP+FP+TN+FN}\ X\ 100$$

Decision Tree

Decision tree is a flow-chart-like tree structure that consists of nodes and branches. (Root node, Internal node and Leaf node)

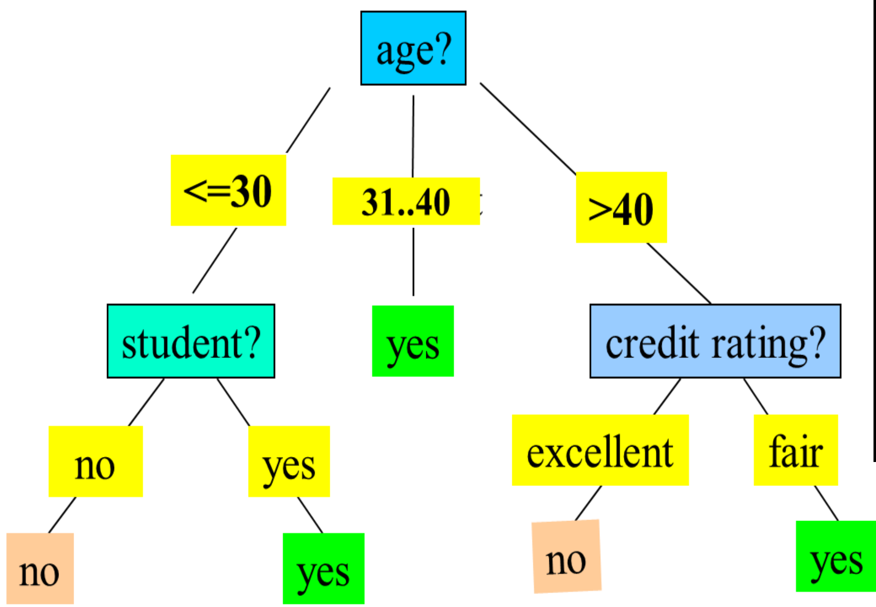
- **Root Node**: The top node of the decision tree with no incoming branch and one or more outgoing branches.
- **Internal Node**(s): has (have) one incoming branch and one or more outgoing branches.
- **Leaf node**: has only one incoming but no outgoing branch and it represents the class label.
- Each internal node and root node denotes an attribute (Feature), each branch represents an outcome of the test.

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - There are no remaining attributes for further partitioning

Decision Tree Induction: An Example

- **Training data set: Buys_computer**
- **The data set follows an example of ID3**
- **Resulting tree:**



age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Decision Tree Example . using ID3

- Extracting Classification Rules from the decision tree

➢ If **Age** (31…40) Then **Buys-Computer** (Yes)

➢ If **Age** (<=30) And **Student** (No) Then **Buys-Computer** (No)

➢ If **Age** (<=30) And **Student** (Yes) Then **Buys-Computer** (Yes)

➢ If **Age** (>40) And **Cr-Rating** (Excellent) Then **Buys-Computer** (No)

➢ If **Age** (>40) And **Cr-Rating** (Fair) Then **Buys-Computer** (Yes)

Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes): $P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$
- This greatly reduces the computation cost: Only counts the class distribution
- If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k [C_i ---class instances]

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

Task:

Classify X using Bayesian classifier ????

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayes Classifier: An Example

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643

P(buys_computer = "no") = 5/14= 0.357

Compute $P(X|C_i)$ for each class

P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222

P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6

P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444

P(income = "medium" | buys_computer = "no") = 2/5 = 0.4

P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667

P(student = "yes" | buys_computer = "no") = 1/5 = 0.2

P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667

P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

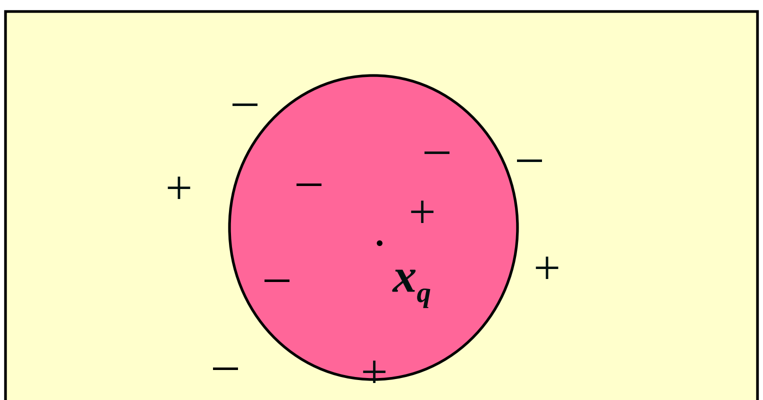
$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028

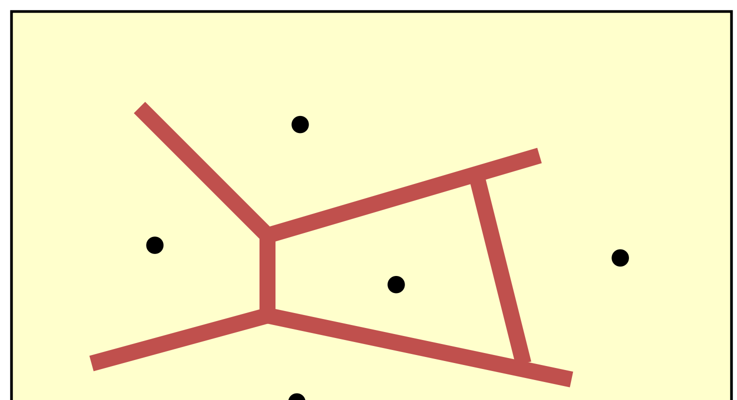
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

The *k*-Nearest Neighbor Algorithm

- All instances correspond to points in the n-D space
- The nearest neighbor are defined in terms of Euclidean distance, dist($\mathbf{X_1}$, $\mathbf{X_2}$)
- Target function could be discrete- or real- valued
- For discrete-valued, *k*-NN returns the most common value among the *k* training examples nearest to x_q
- Vonoroi diagram: the decision surface induced by 1-NN for a typical set of training examples

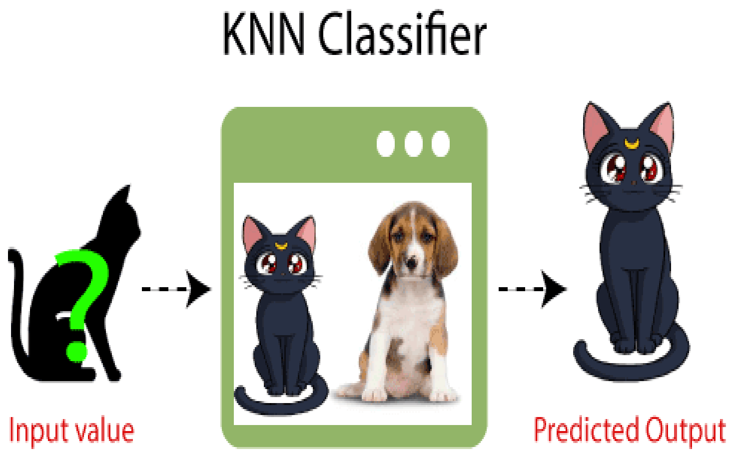


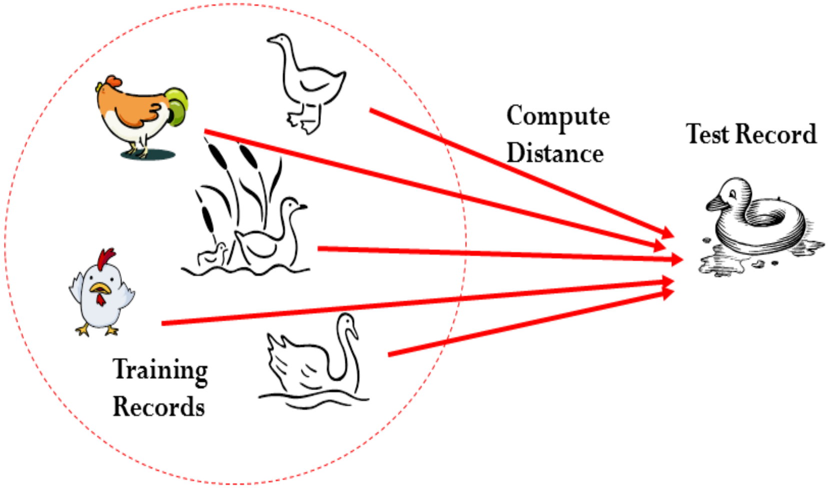


k-Nearest Neighbor (k-NN) Classification

- In k-nearest-neighbor (k-NN) classification, the training dataset is used to classify each member of a “target” dataset.
- There is **no model** created during a learning phase but the training set itself.
- It is called a **lazy-learning** method.
- Basic idea: The basic idea of nearest-neighbor models is that the properties of any particular input *X* are likely to be *similar* to those of points in the neighborhood of *X*.

k-Nearest Neighbor (k-NN) Classification





KNN algorithm works on a similarity measure.

KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Nearest-Neighbor Classifiers

- Requires 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
- To classify an unknown record:
  1) Compute distance to other training records
  2) Identify *k* nearest neighbors
  3) Use class labels of nearest neighbors to determine the class label of unknown record

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

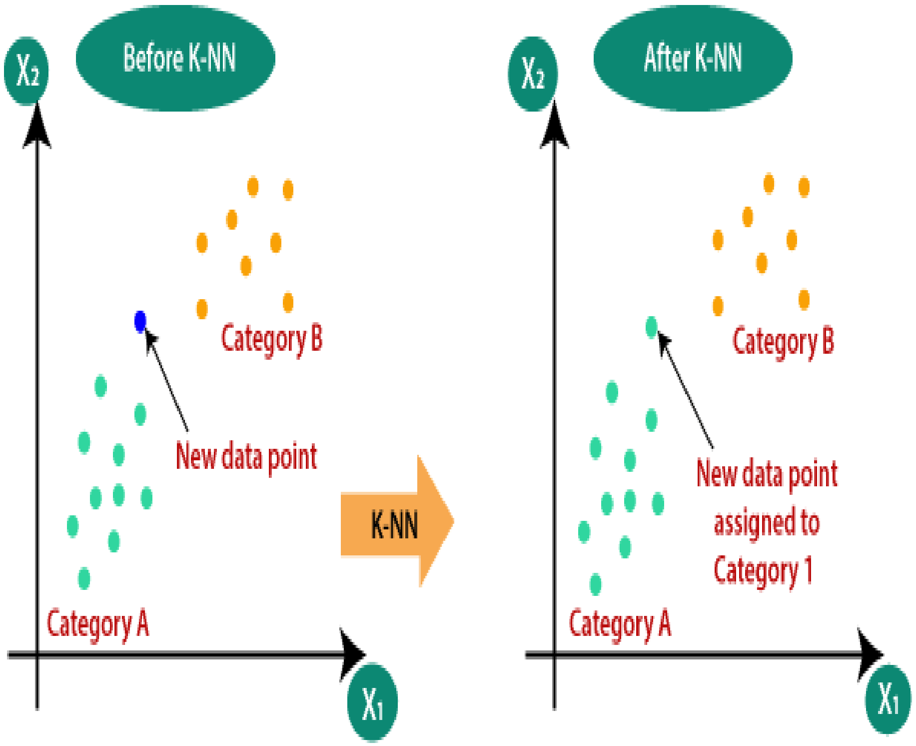
Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

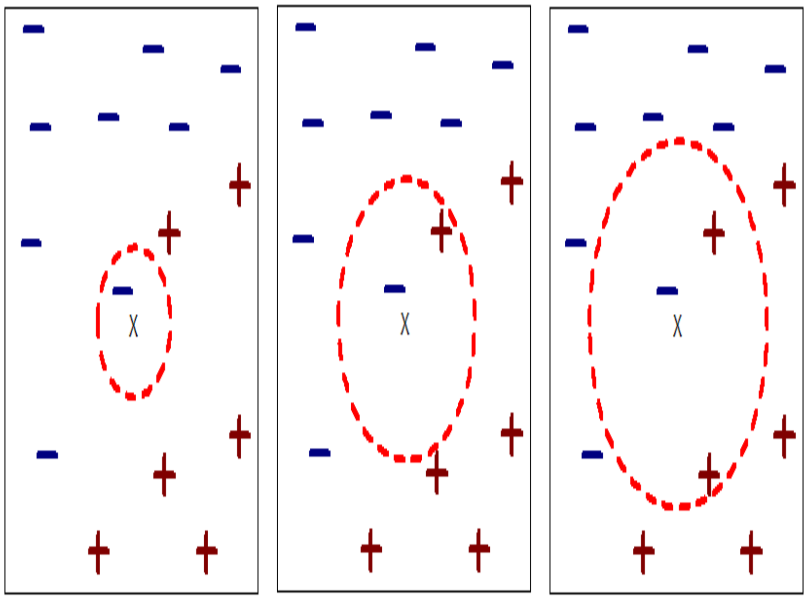
Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

Examples of Nearest Neighbor





(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

ACTIVITY-10(LAB-5)

Investigate any classification problem for a dataset and try to implement those three algorithms i.e. Decision Tree, Naïve Bayes, K-Nearest Neighbors, and analyze the classification accuracy and other performance factors of each type algorithm.

Decision Tree

- R package "party" is used to create decision trees. package "party" has the function ctree() which is used to create and analyze decison tree.

ctree(formula, data)

- formula is a formula describing the predictor and response variables.

- data is the name of the data set used.

Naive Bayes Classifier

- pkgs = c("klaR", "caret", "ElemStatLearn")
- # Install these packages
- # Split the data in training and testing
- # Define a matrix with features, X_train
- # And a vector with class labels, y_train
- # Train the model

train(X_train, y_train, method = 'nb‘)

- # Compute pred using the model to get the predictive accuracy.

K-Nearest Neighbors

- ✓ K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems.
- ✓ ##use knn() function
- ✓ install.packages("e1071")
- ✓ install.packages("caTools")
- ✓ install.packages("class")
- ✓ # Confusiin Matrix

Cheers For the Great Patience!

Query Please?