# Explain the role of machine learning algorithms in Spam filtering. (Unit 1)

* E-mail provides a perfect way to send millions of advertisements at no cost for the sender, and this unfortunate fact is nowadays extensively exploited by several organizations.
* As a result, the e-mailboxes of millions of people get cluttered with all this so-called unsolicited bulk e-mail also known as "spam" or "junk mail".
* Machine learning methods of recent are being used to successfully detect and filter spam emails.
* Different categories of spam filtering techniques that have been widely applied to overcome the problem of email spam.

1. Content Based Filtering Technique : Content based filtering is usually used to create automatic filtering rules and to classify emails using machine learning approaches, such as Naïve Bayesian classification, Support Vector Machine, K Nearest Neighbor, Neural Networks.

This method normally analyses words, the occurrence, and distributions of words and phrases in the content of emails and used then use generated rules to filter the incoming email spams.

1. Case Base Spam Filtering Method : Case base or sample base filtering is one of the popular spam filtering methods. Firstly, all emails both non-spam and spam emails are extracted from each user's email using collection model.

* Subsequently, pre-processing steps are carried out to transform the email using client interface, feature extraction, and selection, grouping of email data, and evaluating the process.
* The data is then classified into two vector sets. Lastly, the machine learning algorithm is used to train datasets and test them to decide whether the incoming mails are spam or non- spam

# Explain the role of machine learning algorithms in Natural Language processing. (Unit 1)

* + The role of machine learning and Al in natural language processing (NLP) and text analytics is to improve, accelerate and automate the underlying text analytics functions and NLP features that turn unstructured text into useable data and insights
  + A machine learning model is the sum of the learning that has been acquired from its training data. The model changes as more learning is acquired.
  + Machine learning for NLP and text analytics involves a set of statistical techniques for identifying parts of speech, entities, sentiment, and other aspects of text.
  + The techniques can be expressed as a model that is then applied to other text, also known as supervised machine learning.
  + It also could be a set of algorithms that work across large sets of data to extract meaning, which is known as unsupervised machine learning.
  + Text data requires a special approach to machine learning. This is because text data can have hundreds of thousands of dimensions (words and phrases) but tends to be very sparse.
  + The most popular NLP machine learning algorithms are:
    - Support Vector Machines
    - Bayesian Networks
    - Maximum Entropy
    - Conditional Random Field

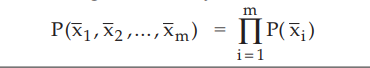
Neural Networks/Deep Learning

# Explain Data formats for supervised learning problem with example. (Unit 1)

* + In a supervised learning problem, there will always be a dataset, defined as a finite set of real vectors with m features each :



* + To consider each X as drawn from a statistical multivariate distribution D. For is purposes, it's also useful to add a very important condition upon the whole dataset X.
  + All samples to be independent and identically distributed (i.i.d). This means all variables belong to the same distribution D, and considering an arbitrary subset of m values, it happens that :

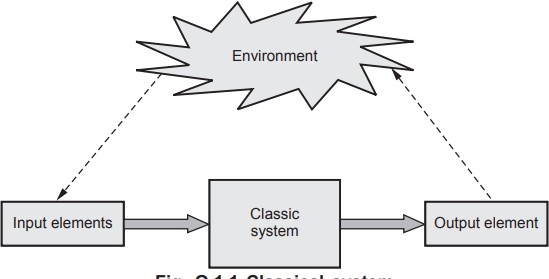


* + The corresponding output values can be both numerical-continuous or categorical. In the first case, the process is called regression, while in the second, it is called classification.
  + Examples of numerical outputs are :



# With reference to machine learning, explain the concept of adaptive machines. (Unit 1)

* Day by day, human uses new tools and machine to simplify their work and reduce the overall effort needed to complete many different tasks.
* Programmable computers are widespread, flexible, and more and more powerful instruments; moreover, the diffusion of the internet allowed us to share software applications and related information with minimal effort.
* Machine learning is a core sub-area of artificial intelligence; it enables computers to get into a mode of self-learning without being explicitly programmed. When exposed to new data, these computer programs are enabled to learn, grow, change, and develop by themselves.
* Machine learning is a method of data analysis that automates analytical model building. It allows computers to find insightful information without being programmed where to look for a particular piece of information; instead, it does this by using algorithms that iteratively learn from data.
* While the concept of machine learning has been around for a long time, the ability to apply complex mathematical calculations to big data automatically, iteratively and quickly has been gaining momentum over the last several years
* Fig. shows a generic representation of a classical system that receives some input values, processes them, and produces output results.



* Adaptive Systems : We can define adaptation as the capacity for a system to change its state in response to some change within its environment.
* An adaptive system then is a system that can change given some external perturbation, and this is done in order to optimize or maintain its condition within an environment by modifying its state.

# Explain role of machine learning the following common un-supervised learning problems: i. Object segmentation ii.Similarity detection

* 1. Object segmentation : Object segmentation is the process of splitting up an object into a collection of smaller fixed-size objects in order to optimize storage and resources usage for large objects. S3 multi-part upload also creates segmented objects, with an object representing each part.
  2. Similarity detection : In contrast to symmetry detection, automatic similarity detection is much harder and more time-consuming. The symmetry factored embedding and the symmetry factored distance can be used to analyze symmetries in points sets. A hierarchical approach was used for building a graph of all subparts of an object.

# What is categorical data? What is its significance in classification problems? (Unit 2)

* That categorical data is defined as variables with a finite set of label values. That most machine learning algorithms require numerical input and output variables. That an integer and one hot encoding is used to convert categorical data to integer data.
* Categorical data is very common in business datasets. For example, users are typically described by country, gender, age group etc., products are often described by product type, manufacturer, seller etc., and so on.
* Several regression and binary classification algorithms are available in scikit-learn. A simple way to extend these algorithms to the multi-class classification case is to use the so-called one-vs-all scheme.
* At learning time, this simply consists in learning one regressor or binary classifier per class. In doing so, one needs to convert multi-class labels to binary labels. LabelBinarizer makes this process easy with the transform method. Machine Learning 2 - 2 Feature Selection
* At prediction time, one assigns the class for which the corresponding model gave the greatest confidence. LabelBinarizer makes this easy with the inverse\_transform method.
* Example :

>>> from sklearn import preprocessing

>>> lb = preprocessing.LabelBinarizer()

>>> lb.fit([1, 2, 6, 4, 2]) LabelBinarizer(neg\_label=0, pos\_label=1,sparse\_output=False)

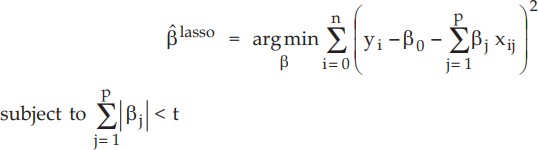
>>> lb.classes\_array([1, 2, 4, 6])

>>> lb.transform([1, 6]) array([[1, 0, 0, 0], [0, 0, 0, 1]])

# Explain the Lasso, and ElasticNet types of regression. (Unit 3)

Lasso

* + One significant problem of ridge regression is that the penalty term will never force any of the coefficients to be exactly zero.
  + Thus, the final model will include all p predictors, which creates a challenge in model interpretation. A more modern machine learning alternative is the lasso.
  + The lasso works in a similar way to ridge regression, except it uses a different penalty term that shrinks some of the coefficients exactly to zero.
  + Lasso : Lasso is a regularized regression machine learning technique that avoids over- fitting of training data and is useful for feature selection.
  + The lasso is a shrinkage method like ridge, with subtle but important differences. The lasso estimate is defined by



* Note the name “lasso" is actually an acronym for : Least Absolute Selection and Shrinkage Operator.
* The only difference from Ridge regression is that the regularization term is in absolute value. But this difference has a huge impact on the trade-off.
* Lasso method overcomes the disadvantage of Ridge regression by not only punishing high values of the coefficients Beta but actually setting them to zero if they are not relevant.

ElasticNet

* ElasticNet, which combines both Lasso and Ridge into a single model with two penalty factors : One proportional to L1 norm and the other to L2 norm.
* Elastic net is a related technique. Use elastic net when you have several highly correlated variables. Lasso provides elastic net regularization when you set the alpha name-value pair to a number strictly between 0 and 1.
* Elastic net can generate reduced models by generating zero-valued coefficients. Empirical studies have suggested that the elastic net technique can outperform lasso on data with highly correlated predictors.
* The elastic net technique solves this regularization problem. For an strictly between 0 and 1 and a nonnegative ƛ , elastic net solves the problem.
* The ElasticNet loss function is defined as :



* The ElasticNet class provides an implementation where the alpha parameter works in conjunction with l1\_ratio. The main peculiarity of ElasticNet is avoiding a selective exclusion of correlated features, thanks to the balanced action of the L1 and L2 norms.

# Write shorts notes on: Bernoulli naiveBayes. (Unit 4)

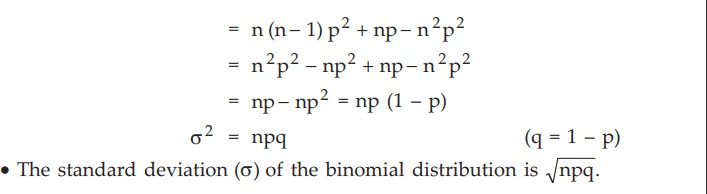
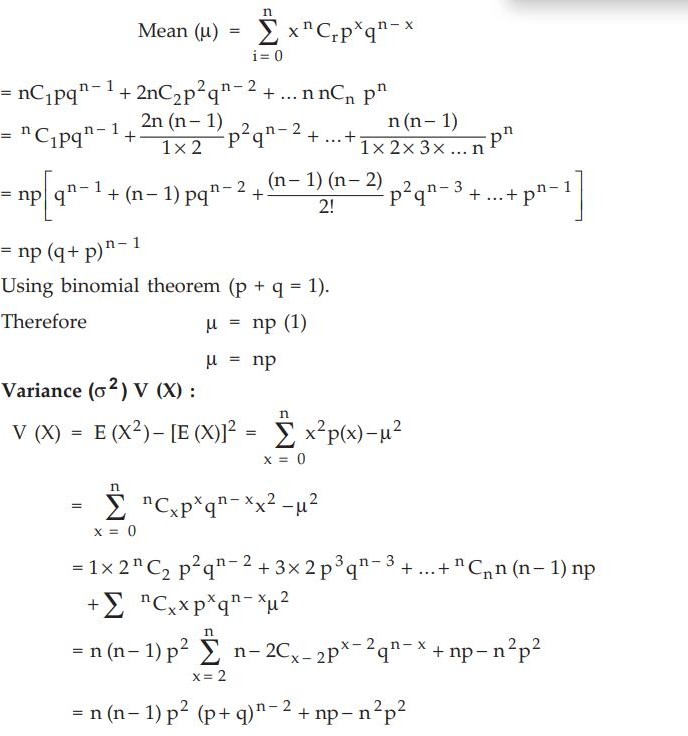
* The Binomial distribution gives the general form of the probability distribution for the random variable r, whether it represents the number of heads in n coin tosses or the number of hypothesis errors in a sample of n examples.

The detailed form of the Binomial distribution depends on the specific sample size n and the specific probability p or errorD (h).

* Binomial distribution applies as follows :
  1. There is a base, or underlying, experiment whose outcome can be described by a random variable (Y). The random variable can take on two possible values.
  2. The probability that Y = 1 on any single trial of the underlying experiment is given by some constant p, independent of the outcome of any other experiment.
  3. A series of n independent trials of the underlying experiment is performed, producing the sequence of independent, identically distributed random variables Y1, Y2, Y3……Yn.

Mean and Variance

* Two properties of a random variable that are often of interest are its expected value (also called its mean value) and its variance. The expected value is the average of the values taken on by repeatedly sampling the random variable



# Write shorts notes on: Multinomial naive Bayes:

# Multinomial distribution is useful to model feature vectors where each value represents, the number of occurrences of a term or its relative frequency.

# If the feature vectors have n elements and each of them can assume k different values with probability Pk, then:

# 

# The sklearn.feature\_extraction module can be used to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image.

# The class DictVectorizer can be used to convert feature arrays represented as lists of standard Python dict objects to the NumPy/SciPy representation used by scikit-learn estimators.

# •DictVectorizer implements what is called one-of-K or "one-hot" coding for categorical features. Categorical features are "attribute-value" pairs where the value is restricted to a list of discrete of possibilities without ordering.

# The DictVectorizer is used when features are stored in dictionaries.

# • Example:

# From sklearn, feature extraction import Dict Vectorizer

# x = [('f1': 'NP', '12': 'in', 13: False, 44: 7).

# (71': 'NP', 12': 'on', 13': True, 14':2},

# ('f1': 'VP', '12': 'in', 'f3': False, 14': 9}]

# vec = DictVectorizer()

# Xe = vec.fit\_transform(X)

# print(Xe.toarray ())

# print(vec.vocabulary\_).

# The result:

# [[1, 0, 1, 0, 0, 7,]

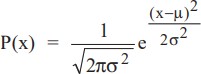
# [1, 0, 0, 1, 1, 2,]

# [0, 1, 1, 0, 0, 9.]

# ['f4': 5, 'f2 = in': 2, f1 = NP': 0, 'f1 = VP': 1, 'f2 on': 3, 'f3': 4)

# Write shorts notes on: Gaussian naive Bayes. (Unit 4)

* + Gaussian naive Bayes is useful when working with continuous values whose probabilities can be modeled using a Gaussian distribution :



* + Conditional Probabilities are also Gaussian Distributed, mean and variance needs to be estimated of each of them using maximum likelihood approach.

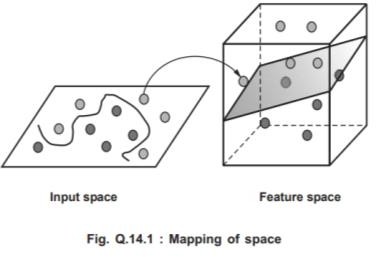
# Naive Bayes

from sklearn.naive\_bayes import GaussianNB clf = GaussianNB()

clf.fit(X\_train, y\_train) pred=clf.predict(X\_test)

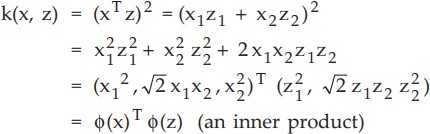
# Explain the non-linear SVM with example. (Unit 4)

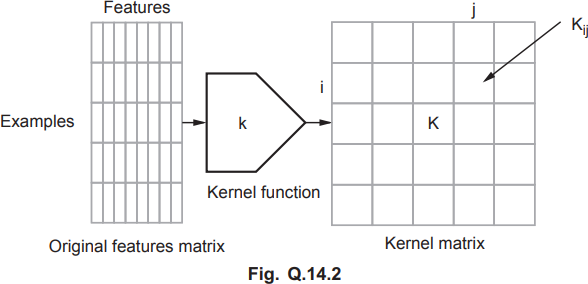
* Kernel methods refer to a family of widely used nonlinear algorithms for machine learning tasks like classification, regression, and feature extraction.
* Any non-linear problem (classification, regression) in the original input space can be converted into linear by making non-linear mapping into a feature space with higher dimension and shown in Fig. Q.14.1



* Often we want to capture nonlinear patterns in the data.

1. Nonlinear Regression : Input-output relationship may not be linear
2. Nonlinear Classification : Classes may not be separable by a linear boundary
   * Kernels : Make linear models work in nonlinear settings.
   * Kernels, using a feature mapping ø, map data to a new space where the original learning problem becomes easy.
   * Consider two data points x = {x1; x2 } and z = {z1; z2 }. Suppose we have a function k which takes as inputs x and z and computes.





* + The above k implicitly defines a mapping ø to a higher dimensional space



* + We didn't need to pre-define/compute the mapping ø to compute k(x, z).
  + The function k is known as the kernel function.
  + K : N × N matrix of pairwise similarities between examples in F space Advantages

1. The kernel defines a similarity measure between two data points and thus allows one to incorporate prior knowledge of the problem domain.
2. Most importantly, the kernel contains all of the information about the relative positions of the inputs in the feature space and the actual learning algorithm is based only on the kernel function and can thus be carried out without explicit use of the feature space.
3. The number of operations required is not necessarily proportional to the number of features.

**#** SVM

from sklearn import svm

>>> X = [[0,0],[2,2]]

>>>y = [0.5,2,5]

>>>clf=svm.SVC(kernel=”poly”).fit(X,y)

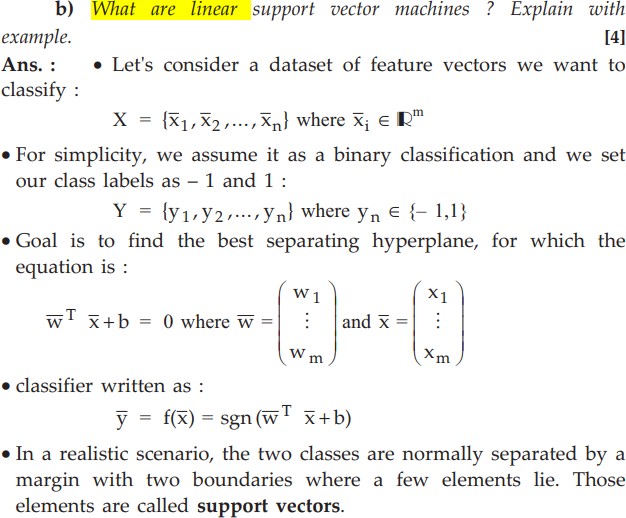
>>>clf.predict([[1,1]])

# Q} What problems are faced by SVM when used with real datasets? (Unit 4)

Main disadvantages:

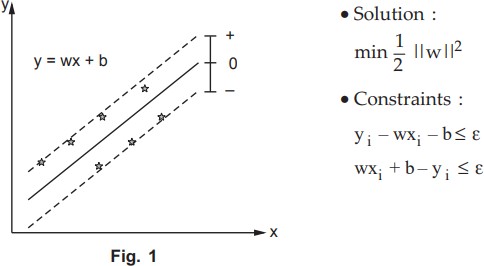
1. Unbalanced data – where the negative instances exceed the positive instances
2. Multilabel classification – SVM is designed for binary classification, multilabel is computationally expensive
   * When large real datasets is used with support vector machine, it can extract a very large number of support vectors to increase accuracy and that can slow down the whole process.
   * To allow finding out a trade-off between precision and number of support vectors, Scikit- learn provides an implementation called NuSVC, where the parameter nu (bounded between 0 and 1) can be used to control at the same time the number of support vectors and training errors.
   * NuSVC is defined as
   * class sklearn.svm.NuSVC(nu=0.5, kernel='rbf', degree=3, gamma=0.0, coef0=0.0, shrinking=True, probability=False, tol=0.001, cache\_size=200)
   * Similar to SVC but uses a parameter to control the number of support vectors. The implementation is based on libsvm.
   * Parameters :
3. nu : float, optional (default=0.5)
   * An upper bound on the fraction of training errors and a lower bound of the fraction of support vectors. Should be in the interval (0, 1].
4. kernel : string, optional (default='rbf')
   * Specifies the kernel type to be used in the algorithm. One of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'. If none is given 'rbf' will be used.
5. degree : int, optional (default=3)
   * degree of kernel function is significant only in poly, rbf, sigmoid
6. gamma : float, optional (default=0.0)
   * Kernel coefficient for rbf and poly, if gamma is 0.0 then 1/n\_features will be taken.
7. coef0 : float, optional (default=0.0)
   * Independent term in kernel function. It is only significant in poly/sigmoid.
8. probability : boolean, optional (default=False) :
   * Whether to enable probability estimates. This must be enabled prior to calling predict\_proba.
9. shrinking : boolean, optional (default=True)

# What are Linear support vector machines? Explain with example (Unit 4)



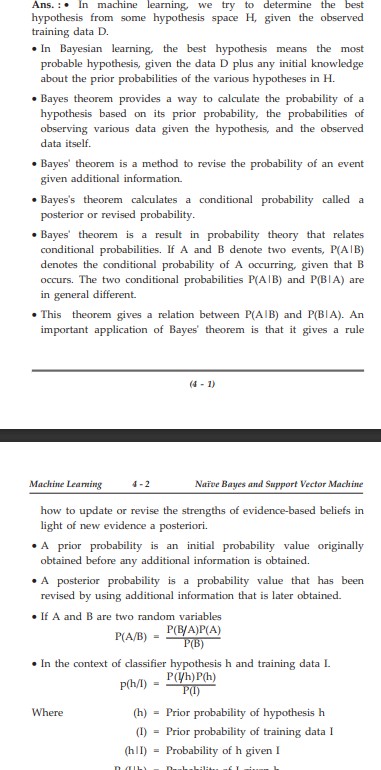
# Explain with example the variant of SVM, the Support vector regression (Unit 4)

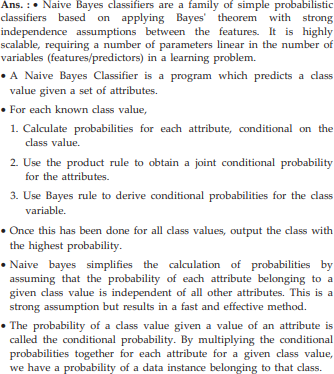
* + Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin).
  + The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences.
  + First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities.
  + In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem



* + But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration.
  + However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.

1. **Define Bayes Theorem. Elaborate Naive Bayes Classifier working with example. (Unit 4)**





# With reference to Deep Learning, Explain the concept of Deep Architectures? (Unit 6)

* + Deep learning architectures are based on a sequence of heterogeneous layers which perform different operations organized in a computational graph.
  + The output of a layer, correctly reshaped, is fed into the following one, until the output, which is normally associated with a loss function to optimize.
  + A fully connected layer is made up of n neurons and each of them receives all the output values coming from the previous layer.
  + It can be characterized by a weight matrix, a bias vector, and an activation function :



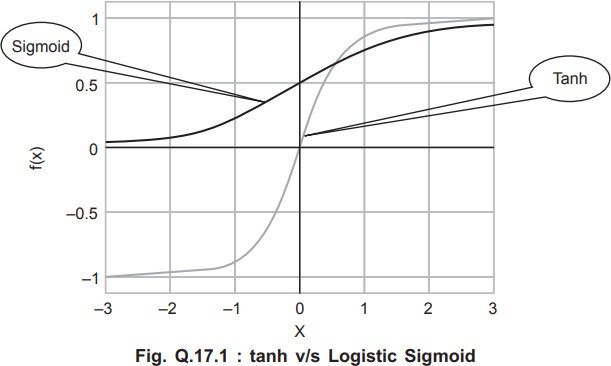
* + They are normally used as intermediate or output layers, in particular when it's necessary to represent a probability distribution.
  + Convolutional layers are normally applied to bidimensional inputs. They are based on the discrete convolution of a small kernel k with a bidimensional input



* + A layer is normally made up of n fixed-size kernels, and their values are considered as weights to learn using a back-propagation algorithm.
  + More than one pooling layer is used to reduced the complexity when the number of convolutions is very high. Their task is to transform each group of input points into a single value using a predefined strategy.
  + A dropout layer is used to prevent overfitting of the network by randomly setting a fixed number of input elements to 0. This layer is adopted during the training phase, but it's normally deactivated during test, validation, and production phases

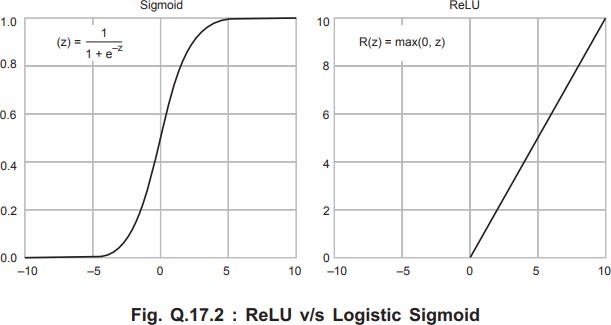
# . What are building blocks of deep networks, elaborate. (Unit 6)

* + The building block of the deep neural networks is called the sigmoid neuron. Deep network also includes neural network, perceptrons, feed forward neural network, Tanh and ReLU neuron.
  + Tanh is also like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). Tanh is also sigmoidal (s - shaped).
  + Fig. Q.17.1 shows tanh v/s Logistic Sigmoid.

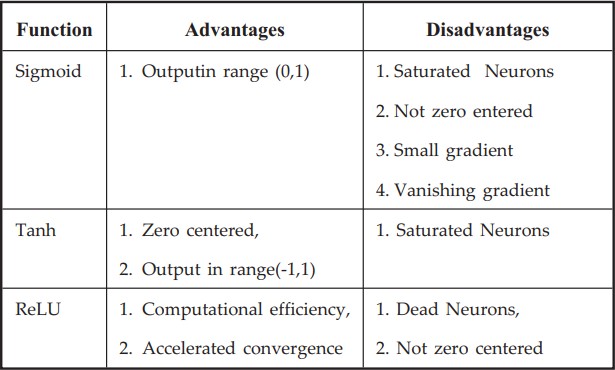


* + Tanh neuron is simply a scaled sigmoid neuron.
  + Problems resolved by Tanh

1. The output is not zero centered
2. Small gradient of sigmoid function
   * ReLU (Rectified Linear Unit) is the most used activation function in the world right now. Since, it is used in almost all the convolution neural networks or deep learning.
   * Fig. Q.17.2 shows ReLU v/s Logistic Sigmoid.



* + As you can see, the ReLU is half rectified (from bottom). f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero.
  + Compared to tanh/sigmoid neurons that involve expensive operations (exponentials, etc.), the ReLUcan be implemented by simply thresholding a matrix of activations at zero.



1. **With reference to Hierarchical Clustering, explain the issue of connectivity constraints. (Unit 6)**

* Scikit-learn also allows specifying a connectivity matrix, which can be used as a constraint when finding the clusters to merge.
* In this way, clusters which are far from each other (nonadjacent in the connectivity matrix) are skipped.
* A very common method for creating such a matrix involves using the k-nearest neighbors graph function, that is based on the number of neighbors a sample has. sklearn.datasets.make\_circles(n\_samples = 100, shuffle=True,noise=None, random\_state=None, factor=0.8)
* It makes a large circle containing a smaller circle in 2d. A simple toy dataset to visualize clustering and classification algorithms.

Parameters :

1. n\_samples : int, optional (default=100)

The total number of points generated. If odd, the inner circle will have one point more than the outer circle.

1. shuffle : bool, optional (default=True) Whether to shuffle the samples.
2. noise : double or None (default=None)

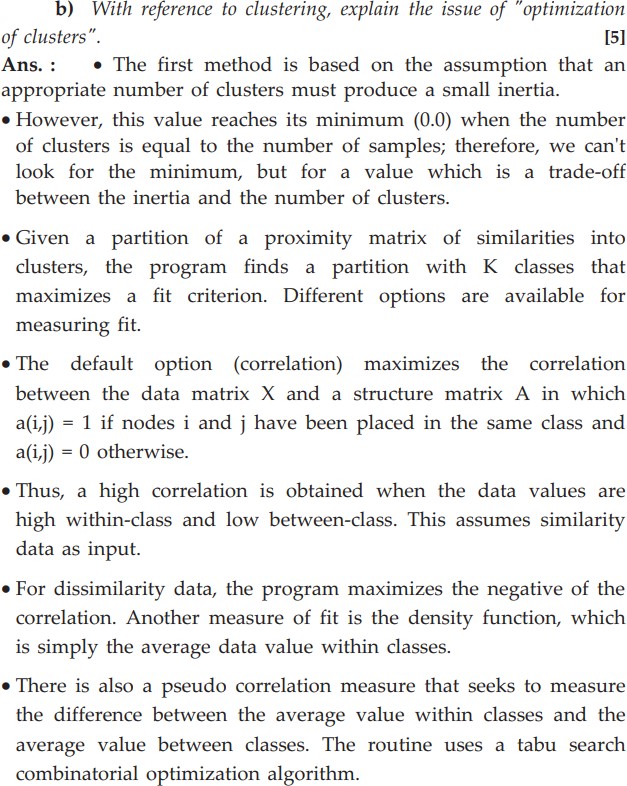
Standard deviation of Gaussian noise added to the data.

1. random\_state : int, RandomState instance or None (default)

Determines random number generation for dataset shuffling and noise. Pass an int for reproducible output across multiple function calls.

1. factor : 0 < double < 1 (default=.8) Scale factor between inner and outer circle

# With reference to Clustering, explain the issue of “Optimization of clusters”. (Unit 5)



# Explain Evaluation methods for clustering algorithms.

Homogeneity

* Homogeneity metric of a cluster labeling given a ground truth. A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class.
* This metric is independent of the absolute values of the labels : a permutation of the class or cluster label values won't change the score value in any way.
* To define the concepts of entropy H(X) and conditional entropy H(X|Y), which measures the uncertainty of X given the knowledge of Y.
* Therefore, if the class set is denoted as C and the cluster set as K, H(C|K) is a measure of the uncertainty in determining the right class after having clustered the dataset.
* To have a homogeneity score, it's necessary to normalize this value considering the initial entropy of the class set H(C) :



* In scikit-learn, there's the built-in function homogeneity\_score() that can be used to compute this value : from sklearn.metrics import homogeneity\_score

Completeness

* A complementary requirement is that each sample belonging to a class is assigned to the same cluster.
* A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster.
* This metric is independent of the absolute values of the labels : a permutation of the class or cluster label values won't change the score value in any way.
* This measure can be determined using the conditional entropy H(K|C), which is the uncertainty in determining the right cluster given the knowledge of the class. Like for the homogeneity score, we need to normalize this using the entropy H(K) :



* We can compute this score (on the same dataset) using the function completeness\_score() :
* from sklearn.metrics import completeness\_score Adjusted Rand Index
* The adjusted rand index measures the similarity between the original class partitioning (Y) and the clustering.
* If total number of samples in the dataset is n, the rand index is defined as :



* Rand index is defined as the number of pairs of objects that are either in the same group or in different groups in both partitions divided by the total number of pairs of objects.
* The Rand index lies between 0 and 1.
* When two partitions agree perfectly, the Rand index achieves the maximum value 1.
* A problem with Rand index is that the expected value of the Rand index between two random partitions is not a constant.
* This problem is corrected by the adjusted Rand index that assumes the generalized hyper- geometric distribution as the model of randomness.
* The adjusted Rand index has the maximum value 1, and its expected value is 0 in the case of random clusters.
* A larger adjusted Rand index means a higher agreement between two partitions. The adjusted Rand index is recommended for measuring agreement even when the partitions compared have different numbers of clusters

**Q} Explain the structure of binary decision tree for a sequential decision process. (Unit 5)**

Ans.: A binary decision tree is a structure based on a sequential decision process.

Starting from the root, a feature is evaluated and one of the two branches is selected. This procedure is repeated until a final leaf is reached, which normally represents the classification target

•Considering other algorithms, decision trees seem to be simpler in their dynamics; however, if the dataset is splittable while keeping an internal balance, the overall process is intuitive and rather fast in its predictions.

Decision trees can work efficiently with unnormalized datasets because their internal structure is not influenced by the values assumed by each feature.

The decision tree always achieves a score close to 1.0, while the logistic regression has an average slightly greater than 0.6.

However, without proper limitations, a decision tree could potentially grow until a single sample is present in every node.

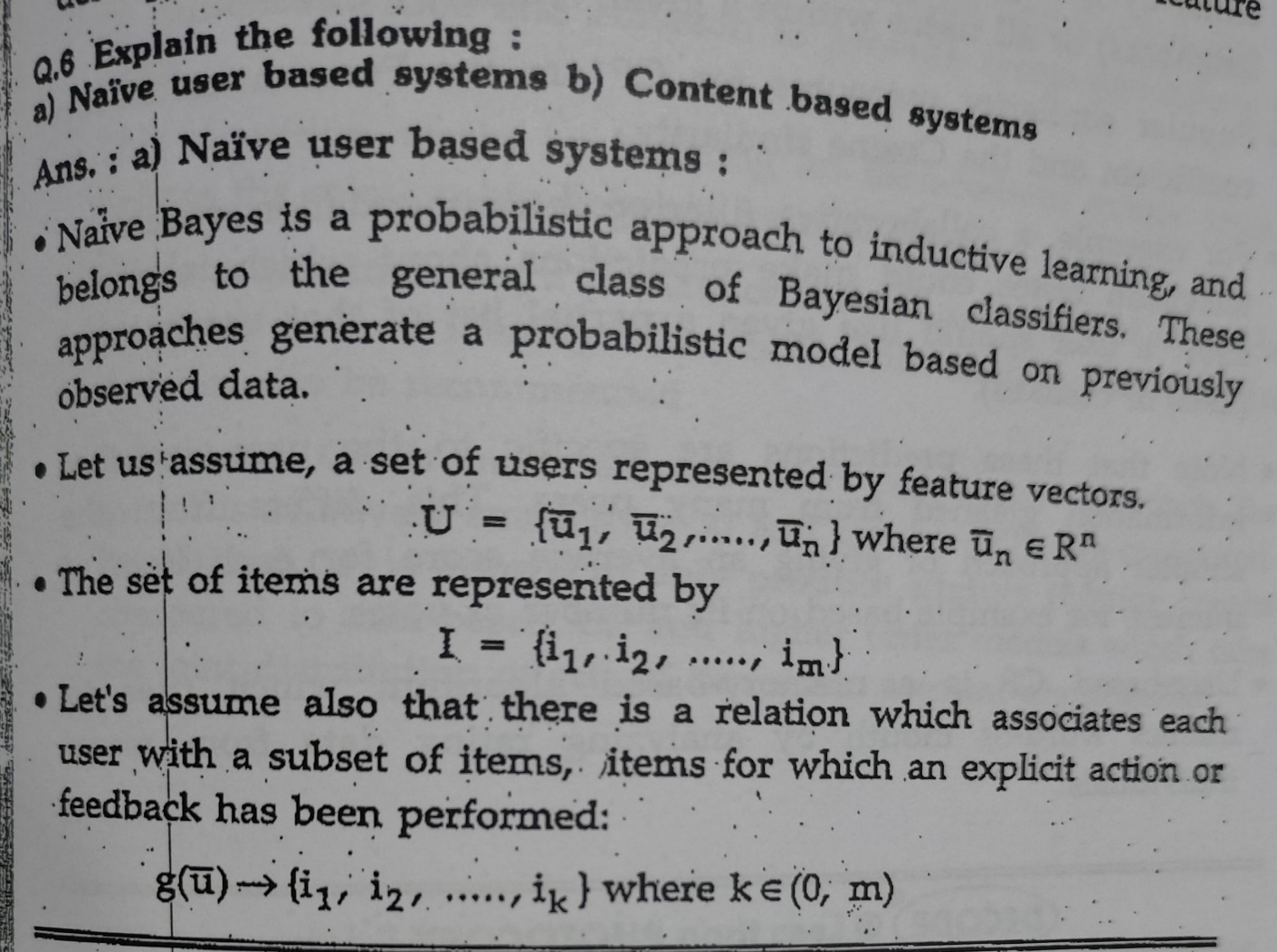
**Q} What is recommendation system?( Unit 6)**

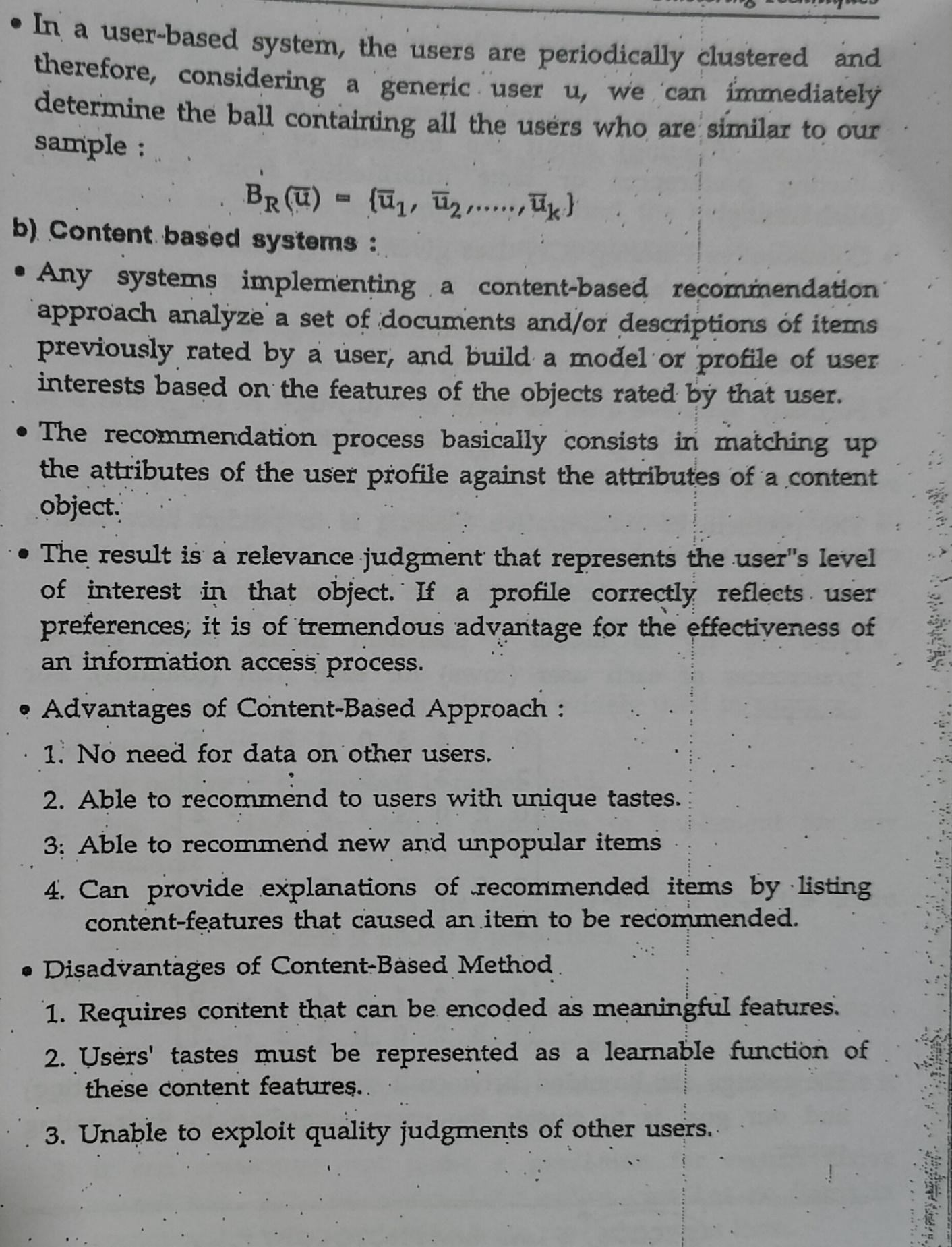
The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read. Item is the general term used to denote what the system recommends to users.

Recommender systems typically produce list a recommendations in one of two ways, through collaborative filtering or through content-based filtering.

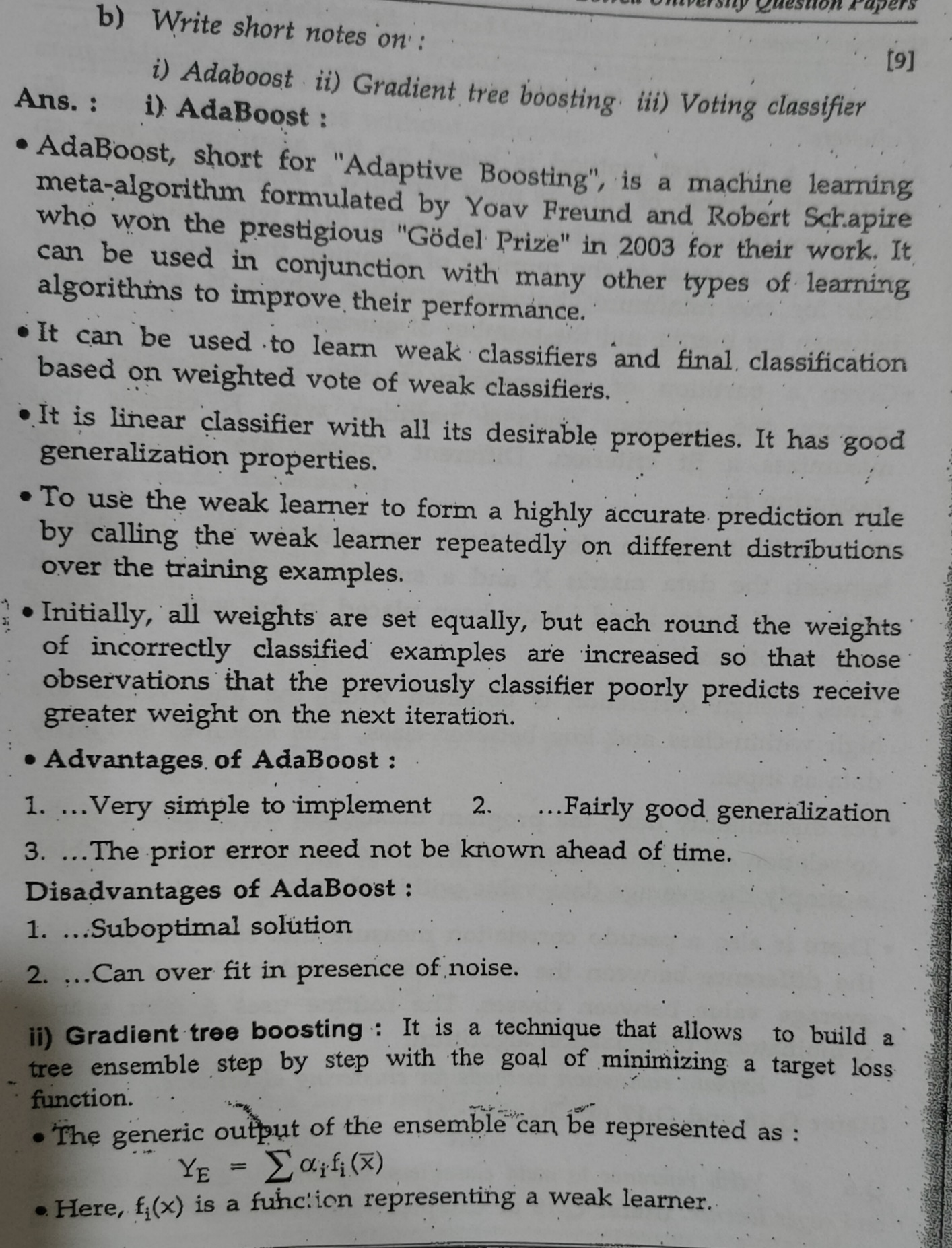
Collaborative filtering systems work by collecting user remark in the form of ratings for items in a given field and exploiting similarities in rating actions amongst several users in determining how to recommend an item. Collaborative filtering systems recommend an item to a user based on opinions of other users.

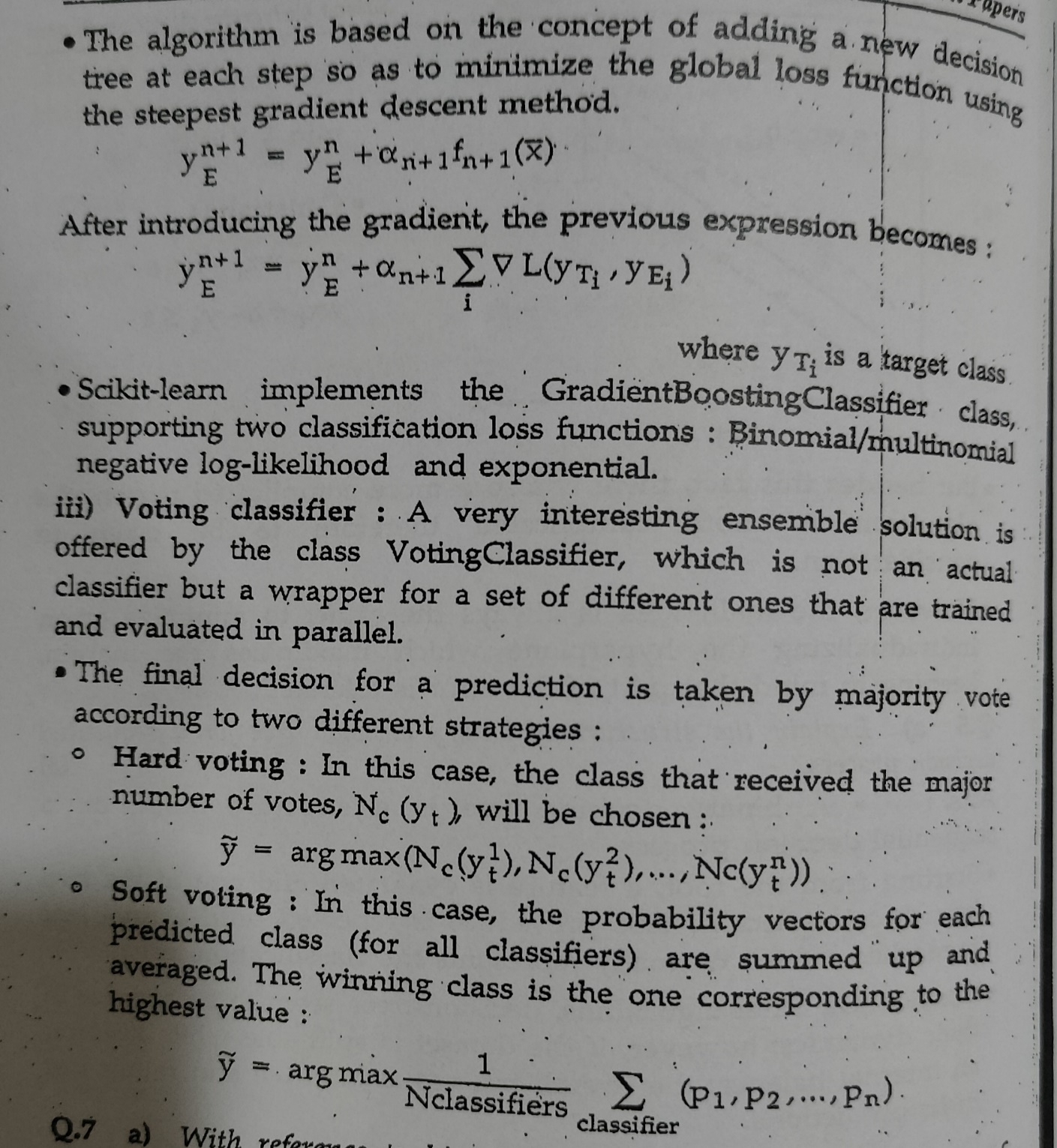
Content-Based Recommending: Recommendations are based on information on the content of items rather than on other users' opinions. Uses a machine learning algorithm to induce a profile of the users preferences from examples based on a feature description of content.





**Q} Write short note on: 1) Adaboost 2) gradient tree boosting 3) voting classifier (Unit 5)**

****

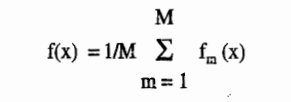
****

**Q} Write notes on : 1) Bagged trees 2) Boosted Trees 3) Random Forest (Unit 6)**

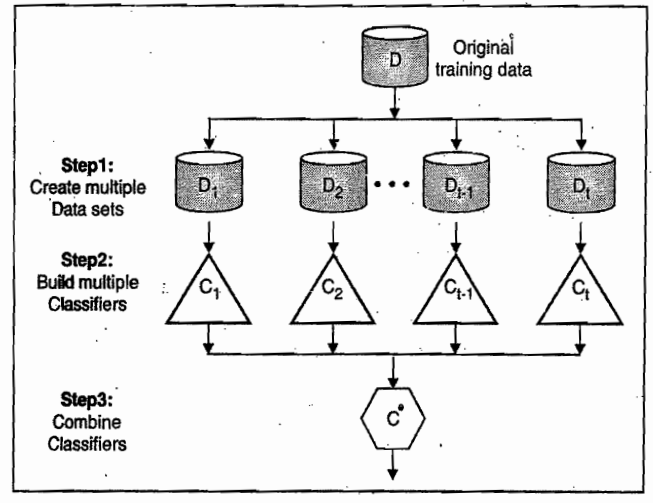
**1) Bagged Trees :-** Base learners are generated in parallel, so it is a parallel ensemble method (eg. Random Forest).

The basic motivation of parallel methods is to exploit independence between the base learners since the error can be reduced dramatically by averaging.

For example, we can train M different trees on different subsets of the data (chosen randomly with replacement) and compute the ensemble:



In generalized bagging, different learners can be used in different population to reduce the variance error.



It averages the prediction from the accumulation of various classifiers used.

A bootstrap method is used, for data set D of n tuples, for each iteration n tuple are sampled with replacement from D.

In every iteration, a classifier model M is learned from training data set Mis

For unknown sample Y, each classifier gives class prediction.

The bagged classifier M uses the voting method i.e. the sample tuple Y is assigned the class with the most votes to tuple Y.

For continuous values, it can be used for prediction by taking the average of all predictions for a given sample.

**2) Boosted Trees :-** It is a family of algorithms that are able to convert weak learners to strong learners.

It is a sequential ensemble method where the base learners are generated sequentially (e.g. AdaBoost).

The basic motivation of sequential methods is to exploit the dependence between the base learners. The overall performance can be boosted by weighing previously mislabelled examples with higher weight.

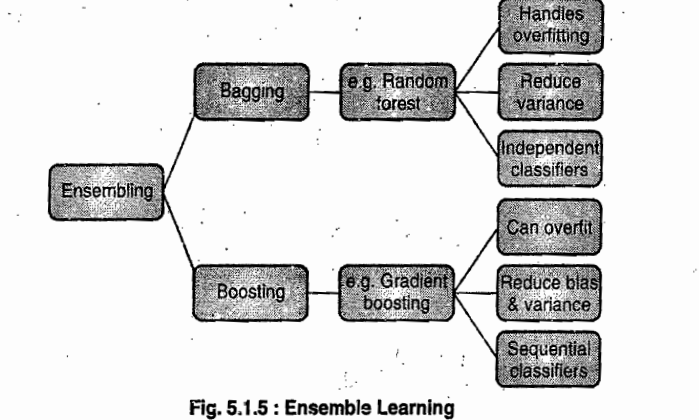
In boosting, each training tuple has weight.

n number of classifiers are learned iteratively.

After learning of Mi classifier, every time the weights are updated for next classifier learning Le. M... So if the tuples which were misclassified by M, will get higher weight for next classifier.

Use voting method, where check the votes of each classifier to get the final M" which helps to get the accuracy. The extended boosting algorithm works for the prediction of continuous values.

Boosting tends to accomplish greater accuracy as compared to bagging, there is a risk of overfitting the model.



**3) Random Forest :-** It is a supervised classification algorithm. It creates the forest with a number of trees as the name suggests

If the number of trees in the forest are more, the accuracy result is high. Random forest handles the missing values.

Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features

The algorithm selects random subset of features to split the node.

Random Forest pseudocode

1. Select features randomly from total "m" features where k << m.

2. Using the bet split polri, calculate the node d from selected k features.

3. Using the best split, split the node d into child nodes.

4 Repeat 1 to 3 steps until "T" number of nodes has been reached.

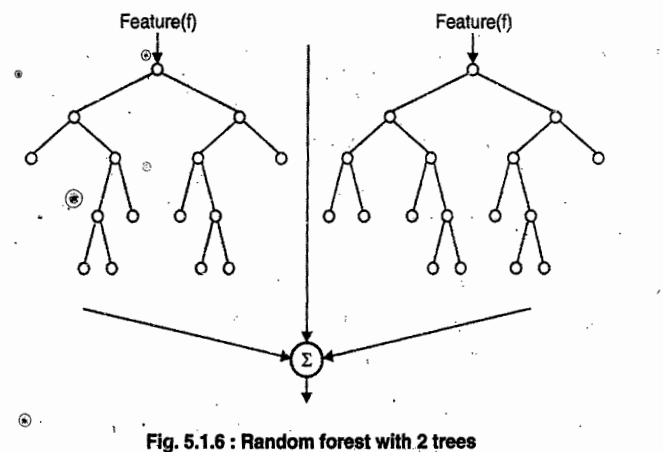
5. Build forest by repeating steps 1 to 4 for "n" number times to create "n" number of trees.

Random forest prediction pseudo code

1. Predict and store the outcome using the rules of each randomly created decision tree.

2. Count on the votes for each predicted target.

3. The last prediction is selected by taking the high voted predicted outcome.

With two trees, you can see how a random forest would look like

Advantages

1. It can be used for both regression and classification tasks.

2. It is a very handy and easy to use algorithm as it's default hyper parameters often produce a good prediction result.

3. The classifier won't overfit the model if there are enough trees in the forest.

Disadvantages

1. Many trees are generated which makes algorithm slow and not suited for real time prediction.

2 It's not a descriptive tool, but only predictive model.

3. If dataset is noisy, then random forest may overfit

4. For data, including categorical variables with different number of levels, random forests are biased favour of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.