Classification & Supervised Learning

5.1 Syllabus topic - Classification: Classification Problem:

- Assigning an object to a certain class based on its similarity to previous examples of other objects
- Can be done with reference to original data or based on a model of that data
- E.g.: Me: "Its round, green, and edible"
- You: "It's an apple
- Classification consists of predicting a certain outcome based on a given input. In order to predict the outcome, the algorithm processes a training set containing a set of attributes and the respective outcome, usually called goal or prediction attribute.
- The algorithm tries to discover relationships between the attributes that would make it possible to predict the outcome. Next the algorithm is given a data set not seen before, called prediction set, which contains the same set of attributes, except for the prediction attribute – not yet known.
- The algorithm analyses the input and produces a prediction. The prediction accuracy defines how "good" the algorithm is. For example, in a medical database the training set would have relevant patient information recorded previously, where the prediction attribute is whether or not the patient had a heart problem. Table 1 below illustrates the training and prediction sets of such database.

Training set

Training set **Age**	**Heart rate**	**Blood pressure**	**Heart problem**
65	78	150/70	Yes
37	83	112/76	No
71	67	108/65	No

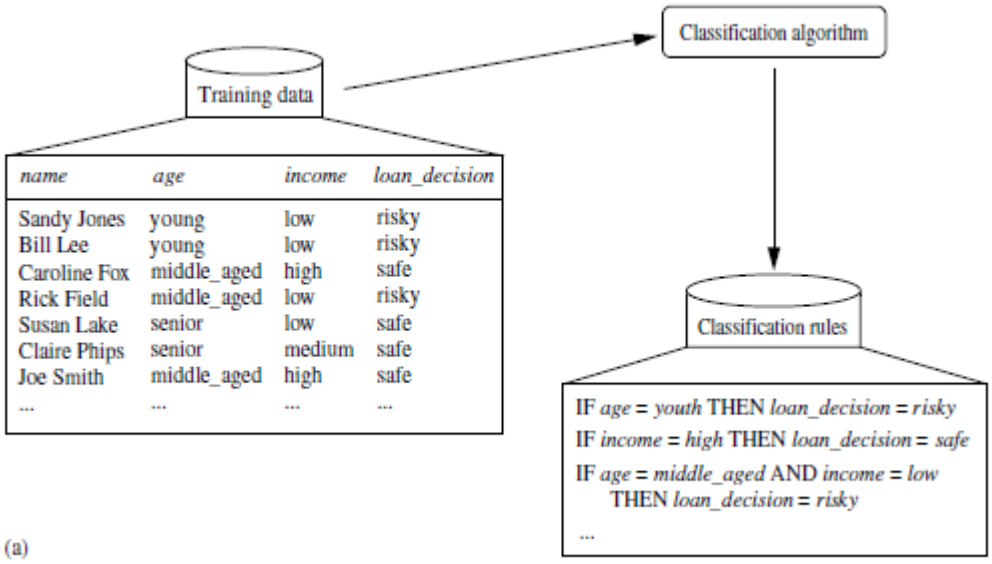
Prediction Set

Prediction set **Age**	**Heart rate**	**Blood pressure**	**Heart problem**
43	98	147/89	?
65	58	106/63	?
84	77	150/65	?

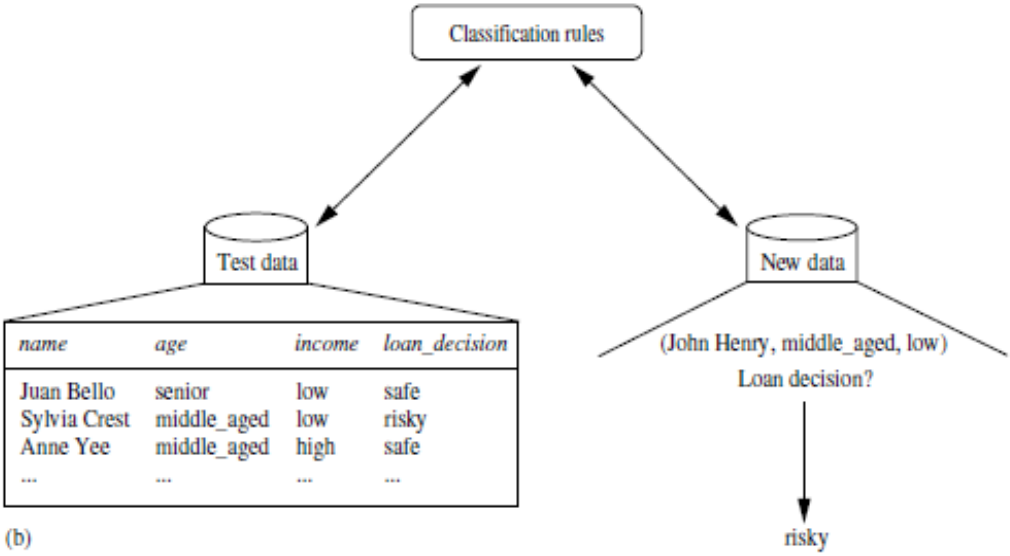
Table 1 – training and prediction sets for medical database

- Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown.
- The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known). "How is the derived model presented?" The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks.
- A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily be converted to classification rules.
- A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units.
- There are many other methods for constructing classification models, such as naïve Bayesian classification, support vector machines, and k-nearest neighbor classification.
- Whereas classification predicts categorical (discrete, unordered) labels, prediction models continuous-valued functions. That is, it is used to predict missing or unavailable numerical data values rather than class labels.
- Although the term prediction may refer to both numeric prediction and class label prediction, in this book we use it to refer primarily to numeric prediction.
- Regression analysis is a statistical methodology that is most often used for numeric prediction, although other methods exist as well.
- Prediction also encompasses the identification of distribution trends based on the available data. Classification and prediction may need to be preceded by relevance analysis, which attempts to identify attributes that do not contribute to the classification or prediction process. These attributes can then be excluded.
- A bank loans officer needs analysis of her data in order to learn which loan applicants are "safe" and which are "risky" for the bank. A marketing manager at All Electronics needs data analysis to help guess whether a customer with a given profile will buy a new computer.
- A medical researcher wants to analyze breast cancer data in order to predict which one of three specific treatments a patient should receive.
- In each of these examples, the data analysis task is classification, where a model or classifier is constructed to predict categorical labels, such as "safe" or "risky" for the loan application data; "yes" or "no" for the marketing data; or "treatment A," "treatment B," or "treatment C" for the medical data.

- These categories can be represented by discrete values, where the ordering among values has no meaning. For example, the values 1, 2, and 3 may be used to represent treatments A, B, and C, where there is no ordering implied among this group of treatment regimes.
- Suppose that the marketing manager would like to predict how much a given customer will spend during a sale at All Electronics. This data analysis task is an example of numeric prediction, where the model constructed predicts a continuous-valued function, or ordered value, as opposed to a categorical label. This model is a predictor.
- Regression analysis is a statistical methodology that is most often used for numeric prediction, hence the two terms are often used synonymously. We do not treat the two terms as synonyms, however, because several other methods can be used for numeric prediction.
- Classification and numeric prediction are the two major types of prediction problems. For simplicity, when there is no ambiguity, we will use the shortened term of prediction to refer to numeric prediction.
- "How does classification work? Data classification is a two-step process, as shown for the loan application data (The data are simplified for illustrative purposes. In reality, we may expect many more attributes to be considered.) In the first step,



(a)



- A classifier is built describing a predetermined set of data classes or concepts. This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels.
- A tuple, X, is represented by an n-dimensional attribute vector,
- X = (x 1 , x 2 , . . . , x n), depicting n measurements made on the tuple from n database attributes, respectively, A 1 , A 2 , . . . , A n . 1 Each tuple, X, is assumed to belong to a predefined class as determined by another database attribute called the class label attribute.
- The class label attribute is discrete-valued and unordered. It is categorical in that each value serves as a category or class.
- The individual tuples making up the training set are referred to as training tuples and are selected from the database under analysis. In the context of classification, data tuples can be referred to as samples, examples, instances, data points, or objects.

5.1.1 General approach to solve classification problem:

- A classification technique (or classifier) is a systematic approach to building classification models from an input data set. Examples include decision tree classifiers, rule-based classifiers, neural networks, support vector machines and naive Bayes classifiers.
- Each technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data.
- The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before.
-

Training set

TID	Attribute1	Attribute1	Attribute1	Class
1	Yes	Large	125k	No
2	No	Medium	1002	No
3	No	Small	70k	No
4	Yes	Medium	120k	No
5	No	Large	95k	Yes
6	No	Medium	60k	No
7	Yes	Large	2230k	No
8	No	Small	85k	Yes
9	No	Medium	75k	No
10	No	Small	90k	Yes

Test set

TID	Attribute1	Attribute1	Attribute1	Class
11	No	Small	55k	?
12	Yes	Medium	80k	?
13	Yes	Large	110k	?
14	No	Small	95k	?
15	No	large	67k	?

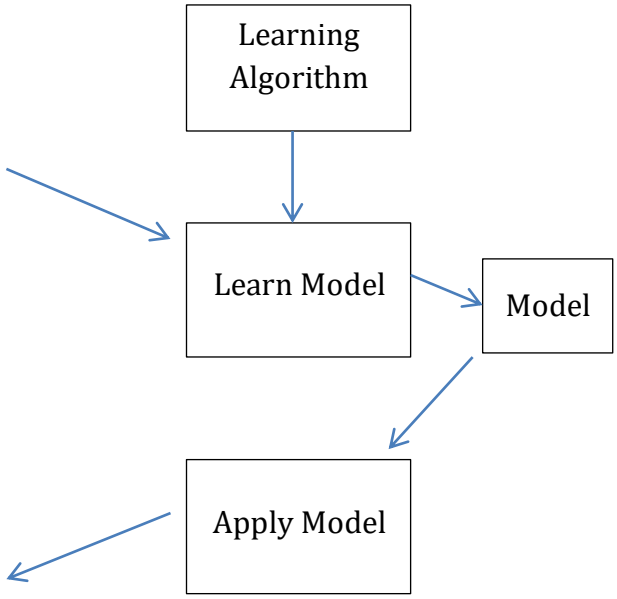


Figure: General approach for building classification model

- Therefore, a ey objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records.
- a general approach for solving classification problems. First, a training set consisting of records whose class labels are known must be provided.
- The training set is used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels.

		Predicted Class	
		Class =1	*Class* =0
Actual Class	*Class* =1	f_{11}	f_{10}
	Class =0	f_{01}	f_{00}

Figure: confusion matrix for a 2-class problem

- Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table known as a confusion matrix. Figure depicts the confusion matrix for a binary classification problem.
- Each entry fij in this table denotes the number of records from class i predicted to be of class j. For instance, f01 is the number of records from class 0 incorrectly predicted as class 1. Based on the entries in the confusion matrix, the total number of correct predictions made by the model is (f11 + f00) and the total number of incorrect predictions is (f10 + f01).

- Although a confusion matrix provides the information needed to determine how well a classification model performs, summarizing this information with a single number would make it more convenient to compare the performance of different models.
- This can be done using a performance metric such as accuracy, which is defined as follows:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{f11 + f00}{f11 + f10 + f01 + f00}$$

- Equivalently, the performance of a model can be expressed in terms of its error rate, which is given by the following equation:

$$Error\ rate = \frac{Number\ of\ wrong\ predictions}{Total\ number\ of\ predictions} = \frac{f10 + f01}{f11 + f10 + f01 + f00}$$

- Most classification algorithms seek models that attain the highest accuracy, or equivalently, the lowest error rate when applied to the test set. We will revisit the topic of model evaluation.

5.2 Syllabus topic - Classification Models:

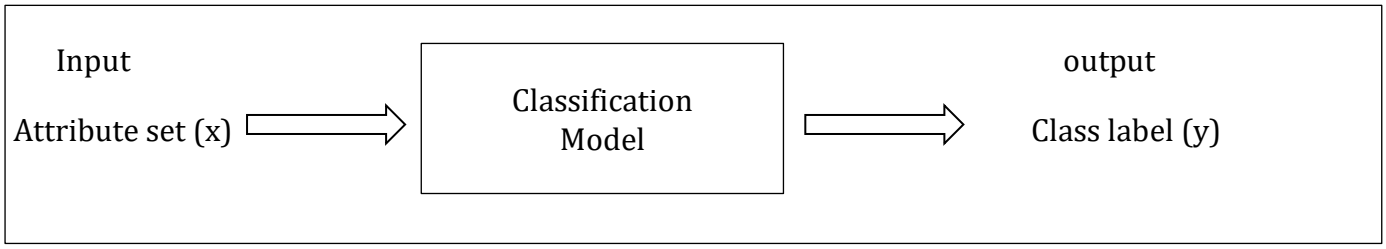


Figure: Classification is the task of mapping an input attribute set x into its class label y

- The input data for a classification task is a collection of records. Each record, also known as an instance or example, is characterized by a tuple (x, y), where x is the attribute set and y is a special attribute, designated as the class label (also known as category or target attribute).
- a sample data set used for classifying vertebrates into one of the following categories: mammal, bird, fish, reptile, or amphibian. The attribute set includes properties of a vertebrate such as its body temperature, skin cover, method of reproduction, ability to fly, and ability to live in water.
- Although the attributes presented in are mostly discrete, the attribute set can also contain continuous features. The class label, on the other hand, must be a discrete attribute. This is a key characteristic that distinguishes classification from regression, a predictive modelling task in which y is a continuous attribute.
- Classification is the task of learning a target function f that maps each attribute set x to one of the predefined class labels y. The target function is also known informally as a classification model.
- A classification model is useful for the following purposes.

  1. Descriptive Modelling

2. Predictive Modelling

1. Descriptive Modelling:

– A classification model can serve as an explanatory tool to distinguish between objects of different classes. For example, it would be useful for both biologists and others to have a descriptive model that summarizes the data and explains what features define a vertebrate as a mammal, reptile, bird, fish, or amphibian.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Figure: The vertebrate data set.

2. Predictive Modelling:

– A classification model can also be used to predict the class label of unknown records. a classification model can be treated as a black box that automatically assigns a class label when presented with the attribute set of an unknown record. Suppose we are given the following characteristics of a creature known as a gila monster:

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
gila monster	cold-blooded	scales	no	No	no	yes	yes	?

– We can use a classification model built from the data set to determine the class to which the creature belongs.

– Classification techniques are most suited for predicting or describing data sets with binary or nominal categories.

- They are less effective for ordinal categories (e.g., to classify a person as a member of high-, medium-, or low- income group) because they do not consider the implicit order among the categories.
- Other forms of relationships, such as the subclass, superclass relationships among categories (e.g., humans and apes are primates, which in turn, is a subclass of mammals) are also ignored.

5.3. Classification Trees:

- A Classification tree labels, records, and assigns variables to discrete classes. A Classification tree can also provide a measure of confidence that the classification is correct.
- A Classification tree is built through a process known as binary recursive partitioning. This is an iterative process of splitting the data into partitions, and then splitting it up further on each of the branches.
- There are two key ideas underlying classification trees.
  1. **Recursive Partitioning:** Space of the independent variables.
  2. **Pruning:** validation data.

1. Recursive Partitioning:

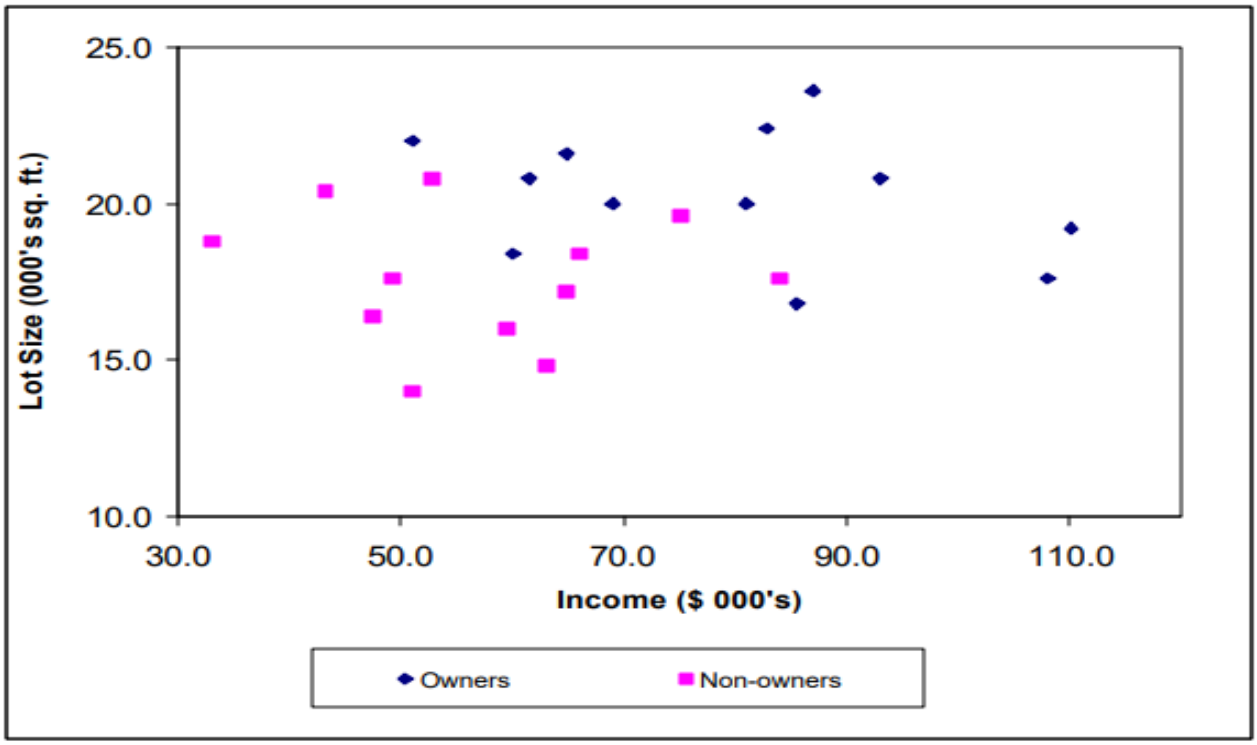
- Let us denote the dependent (categorical) variable by y and the independent variables by x1, x2, x3 …xp.
- Recursive partitioning divides up the p dimensional space of the x variables into non overlapping rectangles.
- This division is accomplished recursively.
- First one of the variables is selected, say xi and a value of xi, say si is chosen to split the p dimensional space into two parts: one part is the p-dimensional hyper-rectangle that contains all the points with xi £ si and the other part is the hyper-rectangle with all the points with xi > ci.
- Then one of these two parts is divided in a similar manner by choosing a variable again (it could be xi or another variable) and a split value for the variable.
- This results in three rectangular regions. (From here onwards we refer to hyper-rectangles simply as rectangles.) This process is continued so that we get smaller and smaller rectangles.
- The idea is to divide the entire x-space up into rectangles such that each rectangle is as homogenous or ‘pure’ as possible. By ‘pure’ we mean containing points that belong to just one class.

Example:

– A riding-mower manufacturer would like to find a way of classifying families in a city into those that are likely to purchase a riding mower and those who are not likely to buy one. A pilot random sample of 12 owners and 12 non-owners in the city is undertaken. The data are shown in Table I and plotted in Figure 1 below. The independent variables here are Income (x1) and Lot Size (x2). The categorical y variable has two classes: owners and non-owners.

Table 1:

Observation	Income ($ 000's)	Lot Size (000's sq. ft.)	Owners=1, Non-owners=2
1	60	18.4	1
2	85.5	16.8	1
3	64.8	21.6	1
4	61.5	20.8	1
5	87	23.6	1
6	110.1	19.2	1
7	108	17.6	1
8	82.8	22.4	1
9	69	20	1
10	93	20.8	1
11	51	22	1
12	81	20	1
13	75	19.6	2
14	52.8	20.8	2
15	64.8	17.2	2
16	43.2	20.4	2
17	84	17.6	2
18	49.2	17.6	2
19	59.4	16	2
20	66	18.4	2
21	47.4	16.4	2
22	33	18.8	2
23	51	14	2
24	63	14.8	2



2. Pruning:

- The second key idea in the CART procedure that of using the validation data to prune back the tree that is grown from the training data using independent validation data was the real innovation, Previously, methods had been developed that were based on the idea of recursive partitioning but they had used rules to prevent the tree from growing excessively and over-fitting the training data.
- For example, CHAID (Chi-Squared Automatic Interaction Detection) is a recursive partitioning method that predates CART by several years and is widely used in database marketing applications to this day.
- It uses a well-known statistical test (the chi-square test for independence) to assess if splitting a node improves the purity by a statistically significant amount.
- If the test does not show a significant improvement the split is not carried out. By contrast, CART uses validation data to prune back the tree that has been deliberately overgrown using the training data.

5.3.1 Bayesian Method:

- We have considered statistical methods which select a single "best" model given the data.This approach can have problems, such as over-fitting when there is not enough data to fully constrain the model fit.

- In contrast, in the "pure" Bayesian approach, as much as possible we only compute distributions over unknowns; we never maximize anything.
- For example, consider a model parameterized by some weight vector w, and some training data D that comprises input-output pairs xi, yi, for i = 1...N. The posterior probability distribution over the parameters, conditioned on the data is, using Bayes' rule, given by

$$p(w|D) = \frac{p(w|D)p(w)}{p(D)}$$

- The reason we want to fit the model in the first place is to allow us to make predictions with future test data. That is, given some future input x new, we want to use the model to predict y new. To accomplish this task through estimation in previous chapters, we used optimization to find ML or MAP estimates of w, e.g., by maximizing
- In a Bayesian approach, rather than estimation a single best value for w, we computer (or approximate) the entire posterior distribution p(w|D). Given the entire distribution, we can still make predictions with the following integral:

$$p(ynew|D, xnew) = \int p(ynew, w|D, xnew)dw$$
$$= \int p(ynew, w|D, xnew)p(w|D, xnew)dw$$

- The first step in this equality follows from the Sum Rule. The second follows from the Product Rule. Additionally, the outputs ynew and training data D are independent conditioned on w, so p(ynew|w,D) = p(ynew|w). That is, given w, we have all available information about making predictions that we could possibly get from the training data D (according to the model). Finally, given D, it is safe to assume that xnew, in itself, provides no information about W. With these assumptions we have the following expression for our predictions:

$$p(ynew|D, xnew) = \int p(ynew|w, xnew)p(w|D)dw$$

- In the case of discrete parameters w, the integral becomes a summation.
- The posterior distribution p(ynew|D, xnew) tells us everything there is to know about our beliefs about the new value ynew. There are many things we can do with this distribution. For example, we could pick the most likely prediction, i.e., argmaxy p(ynew|D, xnew), or we could compute the variance of this distribution to get a sense of how much confidence we have in the prediction.
- We could sample from this distribution in order to visualize the range of models that are plausible for this data.

– The integral in is rarely easy to compute, often involving intractable integrals or exponentially large summations. Thus, Bayesian methods often rely on numerical approximations, such as Monte Carlo sampling; MAP estimation can also be viewed as an approximation.

5.3.2 Association Rule: Structure of Association Rule

– The item or objects in Relational database, transactional databases or other information repositories are considered for finding frequent patterns, association, correlation, or causal structures.
– It searches for interesting relationship among items in a given data set by examining transaction, or shop carts, we can find which items are commonly purchased together. This knowledge can be used in advertising or in goods placement in stores.
– Association rules have the general form

$$I1 \rightarrow (where\ I1 \cap I2 = 0)$$

– Where, I_n are sets of items, for example can be purchased in store.
– The rule should be read as "Given that someone has bought the item in the set I1 they are likely to also buy the items in the set I2".

1. Finding the large Itemsets:

i. The Brute Force Approach:

– Find all association rules
– Calculate the support and confidence for each rule generated in the above step
– The rules that fail the minsup and minconf are pruned for the above list
– The above steps would be a time consuming process, we can have a better approach as given below:

ii. A better approach:

– The Apriori Algorithm

2. Frequent pattern Mining:

Frequent pattern mining is classified in the various ways based on following criteria:

1. Completeness of the pattern to be mined:

– Here we can mine the complete set of frequent Itemsets, closed frequent Itemsets, constrained frequent Itemsets.

2. Levels of Abstraction involved in the rule set:

– Here we use multilevel association rules based on the levels of abstraction of data

3. Number of data dimensions involved in the rule:

- Here we use single dimensional association rule, there is only one dimension or multidimensional association rule if there is more than one dimension.

4. Type of valued handled in the rule:

- Here we use Boolean and quantitative association rules

5. Kind of rule to be mined:

- Here we use association rules and correlation rules based on the kinds of the rules to be mined.

1. Apriori Algorithm:

Apriori Algorithm for finding frequent Itemsets using candidate Generation:

- The Apriori Algorithm solves the frequent item sets problem.
- The Algorithm analyses a data set to determine which combination of items occur together frequently.
- The Apriori algorithm is at the core of various algorithms for data mining problems. The best known problem is finding the association rules that hold in a basket- item relation.

Basic idea:

- An Itemsets can only be a large Itemsets If all its subsets are large Itemsets.
- Frequent itemsets: The sets of items that have minimum support.
- All the subsets of a frequent itemsets must be frequent for e.g (PQ) is a frequent itemsets {P} and {Q} must also be Frequent.
- Find frequent itemsets frequently with cardinality 1 to k (k-itemsets)

Advantages and Disadvantages of Apriori Algorithm:

Advantages:

- The algorithm makes use of large itemsets property.
- The method can be easily parallelized
- The algorithm is easy from implementation point of view

Disadvantages:

- Although the algorithm is easy to implement it needs many database scans which reduces the overall performance.
- Due to Database scans, the algorithm assumes transaction database is memory resident.
-

2. Generating Association Rules from Frequent item sets.

Find the frequent itemsets from transaction database

Using confidence formula generates strong association rules which satisfy both minimum support and minimum confidence.

Support:

- The support of an itemset is the count of that itemset in the total number of transaction, or in other words it is the percentage of the transaction in which the items appear.

If A => B

$$Support\ (A => B) = \frac{\#\_tuples\_containing\_Both\_A\_and\_B}{total\_\#\_of\_tuples}$$

- The support (s) for an association rule X => Y is the percentage of transaction in the database that contains X as well as Y i.e. (X and Y together)
- An itemset is considered to be a large amount if its support is above some threshold called minimum support.

Confidence:

- The confidence or strength for an associate rules A => B is the ratio of the number of transaction that contain A as well as B to the number of transaction that contain A.
- Consider a rule A => B it is measure of ratio of the number of tuples containing both A and B to the number of tuples containing A

$$Cofindence\ (A => B) = \frac{\#\_tuples\_containing\_both\_A\_and\_B}{\#\_tuples\_containing\_A}$$

Solved Examples on Apriori Algorithm:

1. Given the following data apply the Apriori algorithm. Min support =50% Database D.

TID	Itemsets
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Solution:

Step 1: Scan D for count of each candidate. The candidate list is {1,2,3,4,5} and find the support.

C1 =

TID	Itemsets
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

Step 2: Compare candidate support count with minimum support count (i.e.50%)

L1=

TID	Itemsets
{1}	2
{2}	3
{3}	3
{5}	3

Step 3: Generate candidate C2 from L1

C2=

Itemsets
{1,2}
{1,3}
{1,5}
{2 3}
{2,5}
{3,5}

Step 4: Scan D for count of each candidate in C2 and find the support

C2=

TID	Itemsets
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

Step 5: Compare candidate (C2) support count with the minimum support count

C2=

TID	Itemsets
{1,2}	2
{2,3}	2
{2,5}	3
{3,5}	2

Step 6: generate candidate C3 from L2

C3=

Itemsets
{1,3,5}
{2,3,5}
{1,2,3}

Step 7: Scan D for count of each candidate in C3

C3=

TID	Itemsets
{1,3,5}	1
{2,3,5}	2
{1,2,3}	1

Step 8: Compare candidate (C2) support count with the minimum support count

L3=

TID	Itemsets
{2,3,5}	2

Step 9: so data contain the frequent itemset (2, 3, 5)

Therefore the association rules that can be generated from L3 are as shown below with the support and confidence.

Association Rule	Support	Confidence	Confidence %
2^3=>5	2	2/2=1	100%
3^5=>2	2	2/2=1	100%
2^5=>3	2	2/3=0.66	66%
2=>3^5	2	2/3=0.66	66%
3=>2^5	2	2/3=0.66	66%
5=>2^3	2	2/3=0.66	66%

If the minimum confidence threshold is 70% (Given), then only the first and second rules above are output, since these are the only ones generated that are strong.

Final rules are:

Rule1: 2^3=>5

Rule 2: 3^5=>2

Ex.NO 2: A database has five transactions. Let minimum support is 60%.

TID	**Items**
1	Butter, Milk
2	Butter, Dates, Balloon, Eggs
3	Milk, Dates, Balloon, Cake
4	Butter, Milk, Dates, Balloon
5	Butter, Milk, Dates, Cake

Find all the frequent item sets using Apriori algorithm. Show each step

Solution:

Step 1:

Scan database for count of each candidate. The candidate list is {Butter, Milk, Dates, Balloon, Eggs, cake}and find the support.

C1=

Itemset	Support
{Butter}	4
{Milk}	4
{Dates}	4
{Balloon}	3
{Eggs}	1
{cake}	2

Step 2:

Compare candidate support with minimum support (i.e.60%)

Itemset	Support
{Butter}	4
{Milk}	4
{Dates}	4
{Balloon}	3

Step 3:

Generate candidate C2 from L1

Itemset
{Butter, Milk}
{Butter, Dates}
{Butter, Balloon}
{Milk, Dates}
{Milk, Balloon}
{Dates, Balloon}

Step 4:

Scan D for count of each candidate to find the support C2.

Itemset	Support
{Butter, Milk}	3
{Butter, Dates}	3
{Butter, Balloon}	2
{Milk, Dates}	3

{Milk, Balloon}	2
{Dates, Balloon}	3

Step 5:

Compare candidate C2 support count with minimum support count

Itemset	**Support**
{Butter, Milk}	3
{Butter, Dates}	3
{Milk, Dates}	3
{Dates, Balloon}	3

Step 6: Generate candidate C3 from L2

{Balloon, Milk, Dates}

Association rules from this:

Itemset	**Confidence %**
{Milk, Dates}=>{Balloon}	0.67
{Milk, Balloon}=>{Dates}	1.00
{Dates, Balloon}=>{Milk}	0.67
{Balloon}=>{Milk, Dates}	0.67
{Dates}=>{Milk, Balloon}	0.5
{Milk}=>{Dates, Balloon}	0.5

5.4 General Association Rules:

5.5. Clustering:

- Clustering is an unsupervised learning problem in which our goal is to discover "clusters" in the data. A cluster is a collection of data that are similar in some way.
- Clustering is often used for several different problems. For example, a market researcher might want to identify distinct groups of the population with similar preferences and desires.
- When working with documents you might want to find clusters of documents based on the occurrence frequency of certain words. For example, this might allow one to discover financial documents, legal documents, or email from friends.
- Working with image collections you might find clusters of images which are images of people versus images of buildings. Often when we are given large amounts of complicated data we want to look for some underlying structure in the data, which might reflect certain natural kinds within the training data.
- Clustering can also be used to compress data, by replacing all of the elements in a cluster with a single representative element.

1. Types of Clustering:

Broadly speaking, clustering can be divided into two subgroups:

- **Hard Clustering:** In hard clustering, each data point either belongs to a cluster completely or not. For example, in the above example each customer is put into one group out of the 10 groups.
- **Soft Clustering:** In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For example, from the above scenario each costumer is assigned a probability to be in either of 10 clusters of the retail store.

2. Application of Clustering:

Clustering has a large no. of applications spread across various domains. Some of the most popular applications of clustering are:

- Recommendation engines
- Market segmentation
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection

3. Types of Clustering Algorithm:

1. Connectivity models:

- As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away.
- These models can follow two approaches.
- In the first approach, they start with classifying all data points into separate clusters & then aggregating them as the distance decreases.
- In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases. Also, the choice of distance function is subjective.
- These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are hierarchical clustering algorithm and its variants.

2. Centroid models:

- These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters.
- K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters required at the end has to be mentioned beforehand, which makes it important to have prior knowledge of the dataset.
- These models run iteratively to find the local optima.

3. Distribution models:

- These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution (For example: Normal, Gaussian).
- These models often suffer from overfitting. A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.

4. Density Models:

- These models search the data space for areas of varied density of data points in the data space. It isolates various different density regions and assigns the data points within these regions in the same cluster.
- Popular examples of density models are DBSCAN and OPTICS.

5.5.1 Partition Method:

1. K-means Clustering Method:

- We begin with a simple method called K-means. Given N input data vectors {yi}Ni=1, we wish to label each vector as belonging to one of K clusters. This labelling will be done via a binary matrix L, the elements of which are given by

  Li,j = 1 if data point i belongs to cluster j

  0 otherwise

- The clustering is mutually exclusive. Each data vector i can only be assigned P to only cluster: $\sum_{j=1}^{k} Li, j = 1$. Along the way, we will also be estimating a center cj for each cluster. The full objective function for K-means clustering is:

  $$E(c, L) = \sum_{i,j} Li, j||yi - cj||^2$$

- This objective function penalizes the distance between each data point and the center of the cluster to which it is assigned. Hence, to minimize this error, we want to bring the cluster centres close to the data it has been assigned, and we also want to assign the data to nearby centres.
- This objective function cannot be optimized in closed-form, and so an iterative method is required. It includes discrete variables (the labels L), and so gradient-based methods aren't directly applicable.
- Instead, we use a strategy called coordinate descent, in which we alternate between closed-form optimization of one set of variables holding the other variables fixed. That is, we first pick initial values, and then we alternate between updating the labels for the current centres, and then updating the centres for the current labels.
- Here is the K-means algorithm:

pick initial values for $\mathbf{L}$ and $\mathbf{c}_{1:K}$
loop
  // Labeling update: set $\mathbf{L} \leftarrow \arg\min_{\mathbf{L}} E(\mathbf{c}, \mathbf{L})$
  for each data point i **do**
    $j \leftarrow \arg\min_j ||\mathbf{y}_i - \mathbf{c}_j||^2$
    $L_{i,j} = 1$
    $L_{i,a} = 0$ for all $a \neq j$
  end for
  // Centers update: set $\mathbf{c} \leftarrow \arg\min_{\mathbf{c}} E(\mathbf{c}, \mathbf{L})$
  for each center j **do**
    $\mathbf{c}_j \leftarrow \frac{\sum_i L_{i,j} \mathbf{y}_i}{\sum_i L_{i,j}}$
  end for end loop

- Each step of the optimization is guaranteed to lower the objective function until the algorithm converges (you should be able to show that each step is optimal.) However, there is no guarantee

that the algorithm will find the global optimum and indeed it may easily get trapped in a poor local minima.

1. Initialization:

– The algorithm is sensitive to initialization, and poor initialization can sometimes lead to very poor results. Here are a few strategies that can be used to initialize the algorithm:

1. Random labeling:

– Initialize the labeling L randomly, and then run the center-update step to determine the initial centres. This approach is not recommended because the initial centres will likely end up just being very close to the mean of the data.

2. Random initial centers:

– We could try to place initial center locations randomly, e.g., by random sampling in the bounding box of the data. However, it is very likely that some of the canters will fall into empty regions of the feature space, and will therefore be assigned no data. Getting a good initialization this way can be difficult.

3. Random data points as centers:

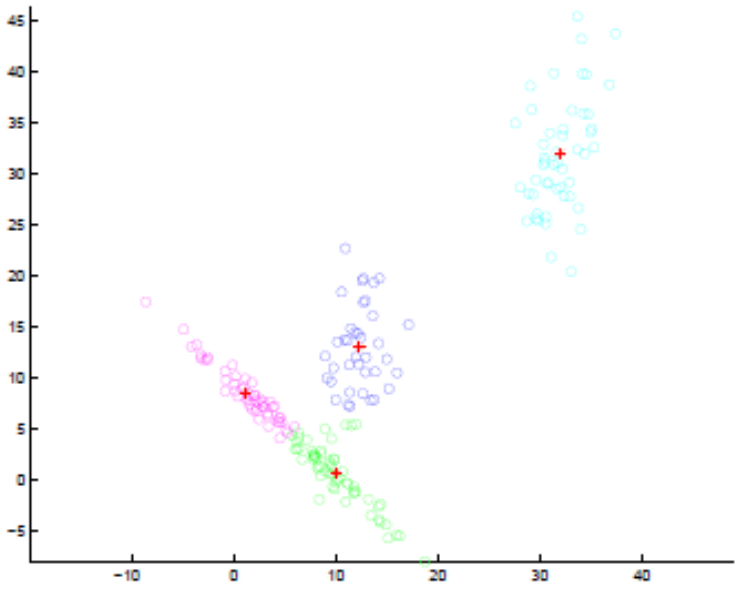
– This method works much better: use a random subset of the data as the initial center locations.

4. K-medoids clustering:

– This will be described below.

5. **Multiple restarts:**

– In multiple restarts, we run K-means multiple times, each time with a different random initialization (using one of the above methods). We then take the best clustering out of all of the runs, based on the value of the objective function.



– K-means applied to a dataset sampled from three Gaussian distributions. Each data assigned to each cluster are drawn as circles with distinct colours. The cluster centers are shown as red stars.
– Another key question is how one chooses the number of clusters, i.e., K. A common approach is to fix K based on some prior knowledge or computational constraints. One can also try different values of K, adding another term to the objective function to penalize model complexity.

Advantages of K-Mean:

- If the variables are large, then K-Means most of the time computationally faster than hierarchical clustering methods.
- K-Means produces tighter clusters than Hierarchical Clustering Method.

Disadvantages of K-Means Partition Algorithm:

- It is difficult to predict the K Value.
- More difficulty in comparing quality of cluster.
- K-Means Algorithm does not work well with global clusters

2. K-Medoid Algorithm:

– It is based on classical partitioning process of clustering The algorithm selects k-medoid initially and then swaps the medoid object with non medoid thereby improving the quality of cluster.

- This method is comparatively robust than K-Mean particularly in the context of _noise‘ or _outlier‘. K-Medoid can be defined as that object of a cluster, instead of taking the mean value of the object in a cluster according to reference point.
- K-Medoids can find the most centrally located point in the given dataset. Procedure of K-Medoid:

Input:

- K:The number of clusters
- D: A data set containing n objects

Output:

- A set of K clusters

Method:

The following steps are

1. The algorithm begins with arbitrary selection of the K objects as medoid points out of n data points(n>K).
2. After selection of the K medoid points, associate each data object in the given data set to most similar medoid.
3. Randomly select non-medoid object O.
4. Compute total cost S of swapping initial medoid object O.
5. If S>0, swap initial medoid with the new one.
6. Repeat steps until there is no change in the medoid.

Advantages of K-Medoid:

1. It is simple to understand and easy to implement.
2. K-Medoid Algorithm is fast and converges in a fixed number of steps.
3. Partition Around Medoid (PAM)algorithm is less sensitive to outliers than other partitioning algorithms.

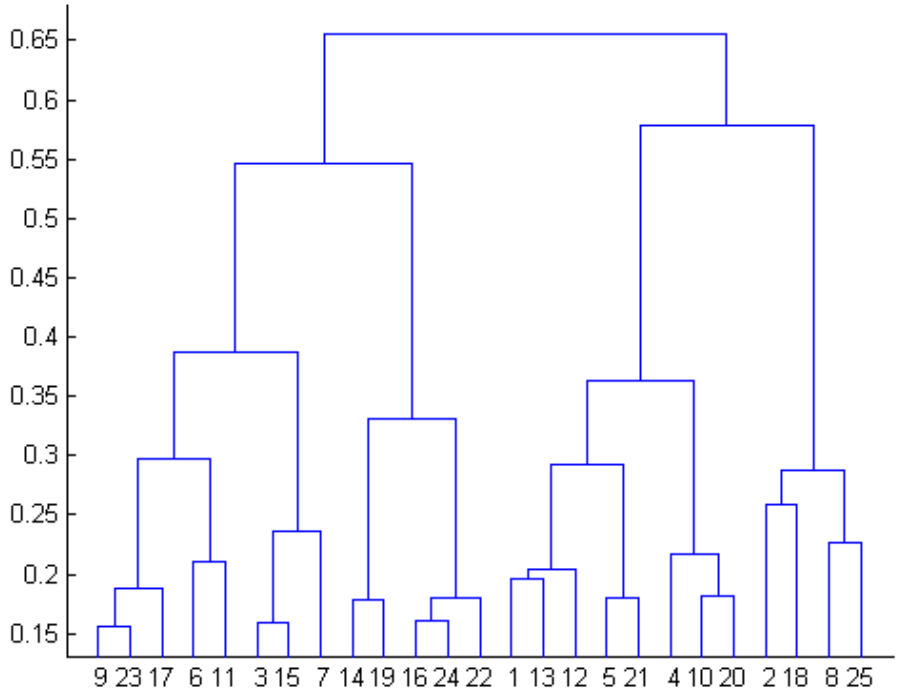
Disadvantages of K-Medoid:

4. 1)K-Mediods is more costly than K-Means Method because of its time complexity.
5. 2) It does not scale well for large datasets. 3)Results and total run time depends upon initial partitions

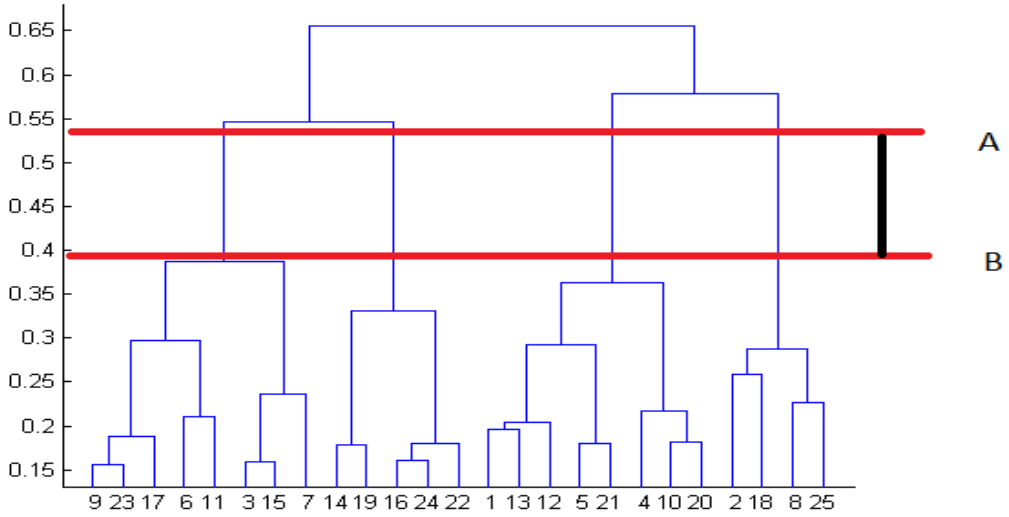
5.5.2. Hierarchical Clustering:

- Hierarchical clustering, as the name suggests is an algorithm that builds hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own.
- Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

- The results of hierarchical clustering can be shown using dendrogram. The dendrogram can be interpreted as:



- At the bottom, we start with 25 data points, each assigned to separate clusters. Two closest clusters are then merged till we have just one cluster at the top.
- The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space.
- The decision of the no. of clusters that can best depict different groups can be chosen by observing the dendrogram.
- The best choice of the no. of clusters is the no. of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster.
- In the above example, the best choice of no. of clusters will be 4 as the red horizontal line in the dendrogram below covers maximum vertical distance AB.



Two important things that you should know about hierarchical clustering are:

- This algorithm has been implemented above using bottom up approach. It is also possible to follow top-down approach starting with all data points assigned in the same cluster and recursively performing splits till each data point is assigned a separate cluster.
- The decision of merging two clusters is taken on the basis of closeness of these clusters. There are multiple metrics for deciding the closeness of two clusters :
  - Euclidean distance: $||a-b||_2 = \sqrt{(\Sigma(a_i-b_i))}$
  - Squared Euclidean distance: $||a-b||_2^2 = \Sigma((a_i-b_i)^2)$
  - Manhattan distance: $||a-b||_1 = \Sigma|a_i-b_i|$
  - Maximum distance:$||a-b||_{INFINITY} = max_i|a_i-b_i|$
  - Mahalanobis distance: $\sqrt{((a-b)^T S^{-1} (-b))}$ {where, s : covariance matrix}

5.5.3 Improving Supervised learning algorithm with Clustering:

– Clustering is an unsupervised machine learning approach, but can it be used to improve the accuracy of supervised machine learning algorithms as well by clustering the data points into similar groups and using these cluster labels as independent variables in the supervised machine learning algorithm?

– Let's check out the impact of clustering on the accuracy of our model for the classification problem using 3000 observations with 100 predictors of stock data to predicting whether the stock will go up or down using R. This dataset contains 100 independent variables from X1 to X100

representing profile of a stock and one outcome variable Y with two levels : 1 for rise in stock price and -1 for drop in stock price.

- The data set is available here :Download it

```
#loading required libraries

library('randomForest')

library('Metrics')

#set random seed

set.seed(101)

#loading dataset

data<-read.csv("train.csv",stringsAsFactors= T)

#checking dimensions of data

dim(data)

## [1] 3000  101

#specifying outcome variable as factor

data$Y<-as.factor(data$Y)

#dividing the dataset into train and test

train<-data[1:2000,]

test<-data[2001:3000,]

#applying randomForest

model_rf<-randomForest(Y~.,data=train)

preds<-predict(object=model_rf,test[,-101])

table(preds)

## preds

##  -1   1

## 453 547
```

```
#checking accuracy
auc(preds,test$Y)
## [1] 0.4522703
```

So, the accuracy we get is 0.45. Now let's create five clusters based on values of independent variables using k-means clustering and reapply random forest.

```
#combing test and train
all<-rbind(train,test)
#creating 5 clusters using K- means clustering
Cluster <- kmeans(all[,-101], 5)
#adding clusters as independent variable to the dataset.
all$cluster<-as.factor(Cluster$cluster)
#dividing the dataset into train and test
train<-all[1:2000,]
test<-all[2001:3000,]
#applying randomforest
model_rf<-randomForest(Y~.,data=train)
preds2<-predict(object=model_rf,test[,-101])
table(preds2)
## preds2
## -1   1
##548 452
auc(preds2,test$Y)
## [1] 0.5345908
```

Final accuracy is poor but clustering has given our model a significant boost from accuracy of 0.45 to slightly above 0.53.