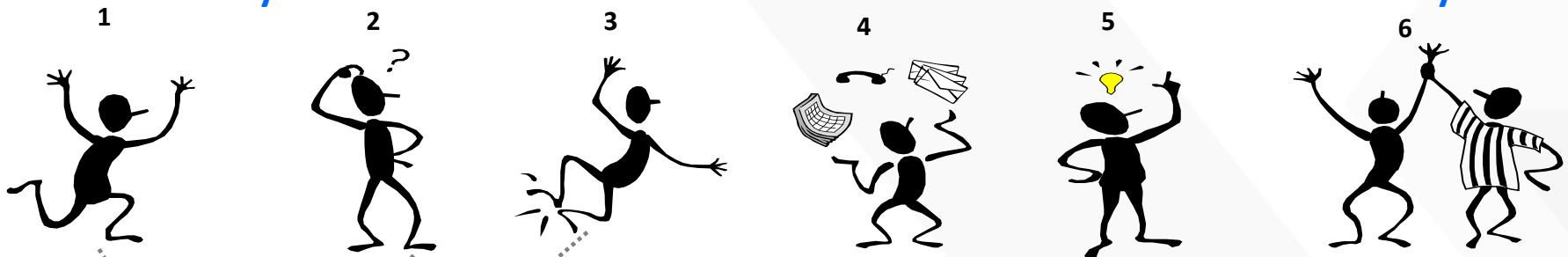
ANALYTIXLABS

Factor Analysis and Segmentation

Introduction to Segmentation

Segmentation

Each individual is so different
that ideally we would want to reach out to each one of them in a different way



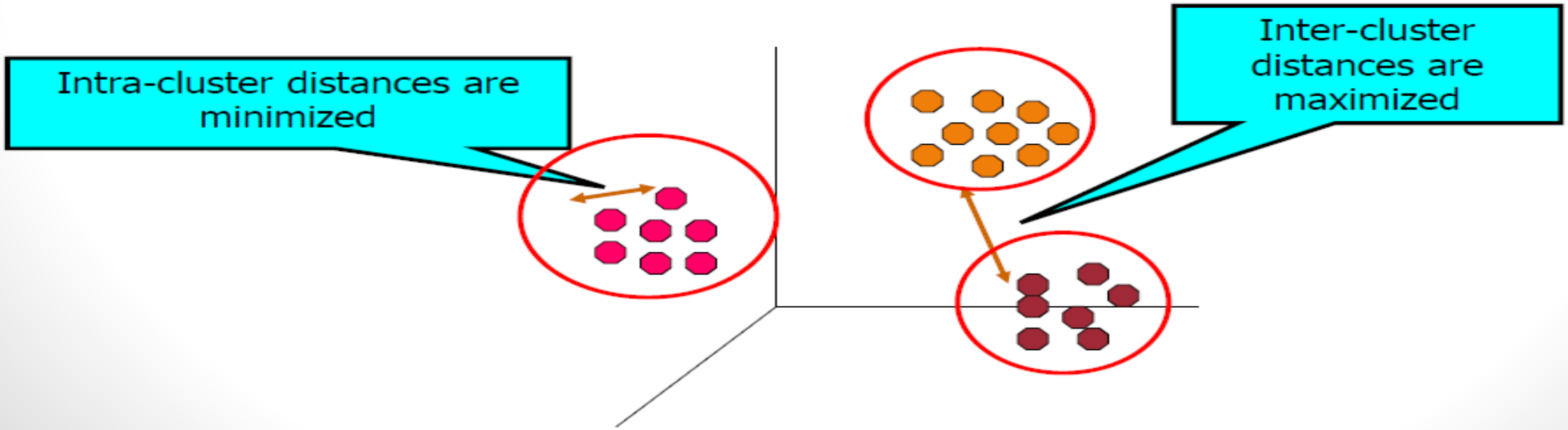
Problem : The volume is too large for customization at individual level

Solution : Identify segments where people have same characters and target each of these segments in a different way

Segmentation is for better targeting

Cluster Analysis

- Cluster is a group of similar objects (cases, points, observations, examples, members, customers, patients, locations, etc)
- Finding the groups of cases/observations/ objects in the population such that the objects are
  - Homogeneous within the group (high intra-class similarity)
  - Heterogeneous between the groups(low inter-class similarity)

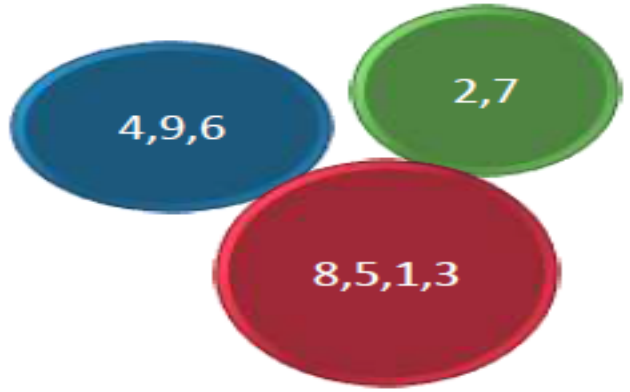


Example

	Maths	Science	Gk	Apt
Student-1	94	82	87	89
Student-2	46	67	33	72
Student-3	98	97	93	100
Student-4	14	5	7	24
Student-5	86	97	95	95
Student-6	34	32	75	66
Student-7	69	44	59	55
Student-8	85	90	96	89
Student-9	24	26	15	22

	Maths	Science	Gk	Apt
Student-1	✓ 94	✓ 82	✓ 87	✓ 89
Student-2	! 46	! 67	✗ 33	✓ 72
Student-3	✓ 98	✓ 97	✓ 93	✓ 100
Student-4	✗ 14	✗ 5	✗ 7	✗ 24
Student-5	✓ 86	✓ 97	✓ 95	✓ 95
Student-6	✗ 34	✗ 32	✓ 75	! 66
Student-7	✓ 69	! 44	! 59	! 55
Student-8	✓ 85	✓ 90	✓ 96	✓ 89
Student-9	✗ 24	✗ 26	✗ 15	✗ 22

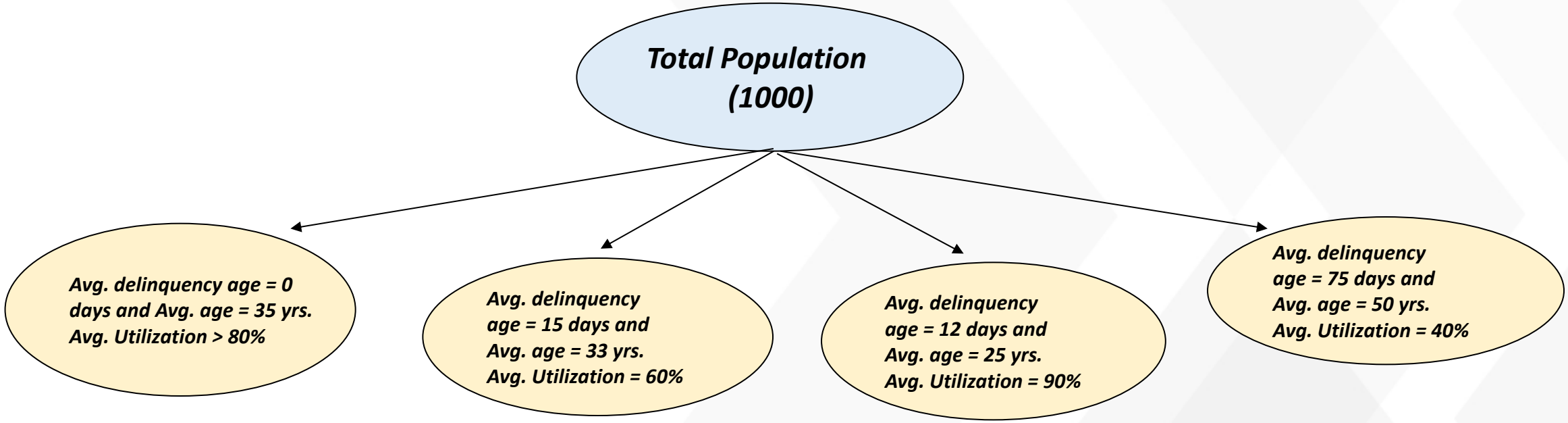
	Maths	Science	Gk	Apt
Student-4	✗ 14	✗ 5	✗ 7	✗ 24
Student-9	✗ 24	✗ 26	✗ 15	✗ 22
Student-6	✗ 34	✗ 32	✓ 75	! 66
Student-2	! 46	! 67	✗ 33	✓ 72
Student-7	✓ 69	! 44	! 59	! 55
Student-8	✓ 85	✓ 90	✓ 96	✓ 89
Student-5	✓ 86	✓ 97	✓ 95	✓ 95
Student-1	✓ 94	✓ 82	✓ 87	✓ 89
Student-3	✓ 98	✓ 97	✓ 93	✓ 100



Business Example

Consider a portfolio with 1000 customers having Credits. Business wants to make different strategies to different groups of people. How company can group them into similar groups?

In this case we need some profiling as below: -



We can exclude the group with avg. delinquency age = 75 days from mailing

This type of segmentation is known as 'Subjective Segmentation'. It gives the salient characteristics of the best customers

Applications of Segmentation

- **Market Segmentation:** Grouping people (with the willingness, purchasing power, and the authority to buy) according to their similarity in several dimensions related to a product under consideration.
- **Sales Segmentation**: Clustering can tell you what types of customers buy what products
- **Credit Risk**: Segmentation of customers based on their credit history
- **Operations**: High performer segmentation & promotions based on person's performance
- **Insurance**: Identifying groups of motor insurance policy holders with a high average claim cost.
- **City-planning:** Identifying groups of houses according to their house type, value, and geographical location
- **Geographical**: Identification of areas of similar land use in an earth observation database.

Customer Segmentation

Customer Segmentation:

- Customer segmentation is the process of splitting your customer database into smaller groups. By focusing on specific customer types, you will maximize customer lifetime value and better understand who they are and what they need.

Typically customers differ in terms of:

- Products they are interested in
- Marketing channels they interact with (e.g. offline media like TV and press, social networks etc.)
- The maximum amount they can pay for a product (willingness to pay)
- Types of promotions and benefits they expect (discounts, free shipping)
- Buying patterns and frequency.

Key variables to use for customer segmentation

Geographical location – knowing where customers live can give you a good idea on their income and lifestyle (you can also incorporate databases like Experian Mosaic)

Age and gender – younger customers are often more impulsive and frequent buyers while female customers might have a higher long-term value

Acquisition channel – e.g. customers from Social Media are often less valuable then customers navigating to your site directly

First product purchased – pay close attention to the transaction value and product category to differentiate between price-focused and quality-focused customers

Device types – e.g. customers using a mobile device typically spend less than customers on a desktop PC

Recency, Frequency and Monetary value of customer transactions is a complete segmentation strategy

etc…

Applications of customer segmentation

Customer segmentation can help other parts of your business. It will allow you to:

- ✓ **Improve customer retention** by providing products tailored for specific segments
- ✓ **Increase profits** by leveraging disposable incomes and willingness to spend
- ✓ **Grow you business quicker** by focusing marketing campaigns on segments with higher propensity to buy
- ✓ **Improve customer lifetime value** by identifying purchasing patterns and targeting customers when they are in the market
- ✓ **Retain customers** by appearing as relevant and responsive
- ✓ **Identify new product opportunities** and improve the products you already have
- ✓ **Optimize operations** by focusing on geographies, age groups etc. with the most value
- ✓ **Increase sales** by offering free shipping to high frequency buyers
- ✓ **Offer improved customer support** to VIP customers
- ✓ **Gain brand evangelists** by incentivising them to comment, review or talk about your product with free gifts or discounts
- ✓ **Reactivate customers** who have churned and no longer interact with you

Types of customer Segmentation

- ✓ **Value Based Segmentation: Customer ranking and segmentation according to current and expected/estimated customer value**
- ✓ **Life Stage Segmentation: Segmentation according to current life stage which he/she belongs**
- ✓ **Loyalty Segmentation: Segmentation according to current & Previous value**
- ✓ **Behavioral Segmentation: Customer segmentation based on behavioral attributes**

There are 3 approaches to behavioral segmentation

Behavioral segmentation

	Description	When to do	Suggested technique	Client example
1 **Rule-based:** Hypothesis driven	Segment customers, **manually**, based on **1 to 3 factors** to drive specific business objective	▪ Only **a couple factors** are thought to drive the segments ▪ **Known hypothesis** to cut the data to create segments	Cross-tabs and conditional data cuts	▪ **Cable client** segmented prospects on their potential telecom spend and used the segmentation to align sales resources and offers to improve go-to-market strategy
2 **Supervised:** With a dependent variable	Segment customers using **predictive algorithm**, based on **high number of factors** that potentially drive a specific outcome	▪ **Data-driven** segments desired, but first and foremost segments need to be **differentiated on a specific outcome/metric** (e.g. revenue)	CHAID	▪ **Telecom client** segmented customers on various factors that drive churn propensity and targeted high churn segments with retention campaigns and offers
3 **Unsupervis-ed:** Without a dependent variable	Segment customers using **clustering algorithm** based on **high number of factors**	▪ **Data-driven** segments desired ▪ Segments need to be differentiated across **many behavioral factors**	TwoStep, K-Means	▪ **Retail client** segmented customers on behavioral shopping factors that included category spend, shopping frequency/tendency, and store/channel shopped to inform merchandising and offer strategy

RFM SEGMENTATION

RFM stands for **Recency**, **Frequency** and **Monetary**

- It is the easiest form of customer database segmentation
- Often used for reactivation campaigns, high valued customer programs, combating churn etc.

RFM Metrics:

RECENCY
The *freshness* of customer activity.

e.g. time since last activity

FREQUENCY
The *frequency* of customer transactions.

e.g. the total number of recorded transactions

MONETARY
The *willingness* to spend.

e.g. the total transaction value

RFM Segmentation can be applied to **activity**-related data that has **measurable value** and is **repeatable**

E-COMMERCE
Orders, visits

SOCIAL MEDIA
Sharing, liking, engagement

GAMING
In-app purchases, levels played

DISCUSION BOARDS
Posting, up-votes

LEAD MANAGEMENT
Engagement, value

You can use **more than one RFM segmentation**

Purchase history
Website visits
Social engagement

RFM Metrics can have multiple definitions

TOTALS
- **R:** Time since last transaction
- **F:** Total number of transactions
- **M:** Total transactions value

*Transactions can only **increase** customer value in the segmentation*

Easy to explain

AVERAGES
- **R:** Time since last transaction
- **F:** Average time between transactions
- **M:** Average transaction value

*Transactions can both **increase and decrease** customer value in the segmentation*

Complicates campaigning

RFM SEGMENTATION- STEPS

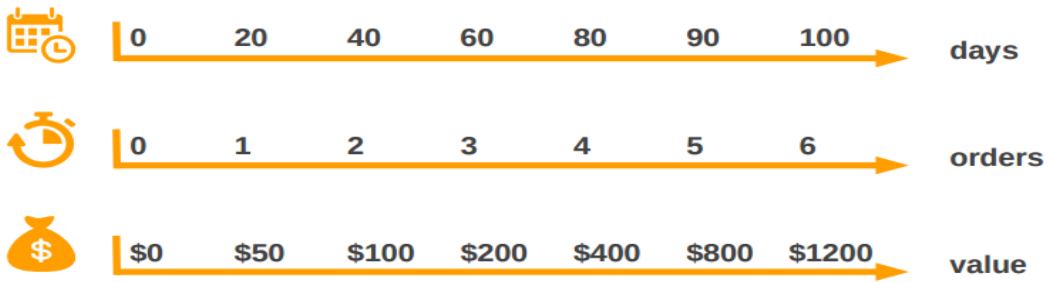
Step 1: Calculate the RFM metrics for each customer

Customer	Recency	Frequency	Monetary
A	53 days	3 transactions	$230
B	120 days	10 transactions	$900
C	10 days	1 transaction	$20

This is called the **RFM Table**

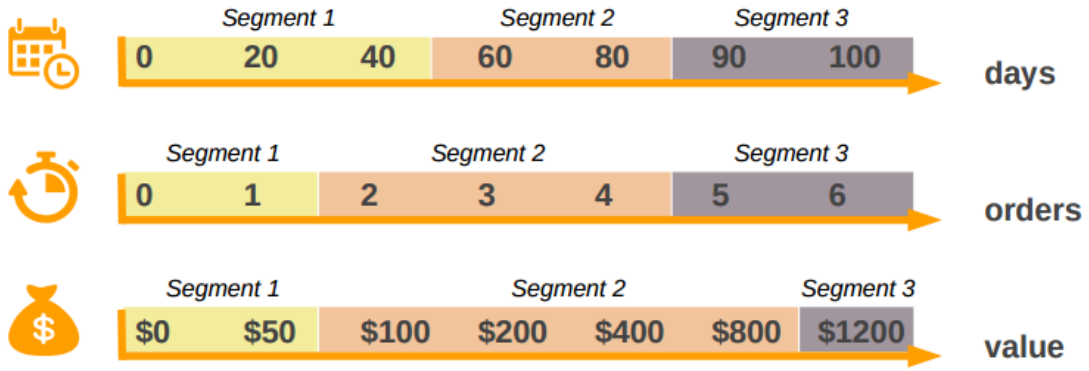
… and can be easily computed in SQL, R, Spark etc.

Step 2: Find the distribution for each metric…



…and define the segmentation…

… by splitting values into bins.



The easiest way to split metrics into segments is by using quantiles:

- This gives a starting point for detailed analysis
- 4 segments are easy to grasp and action

There are much better ways to choose segmentation points!

RFM SEGMENTATION - STEPS

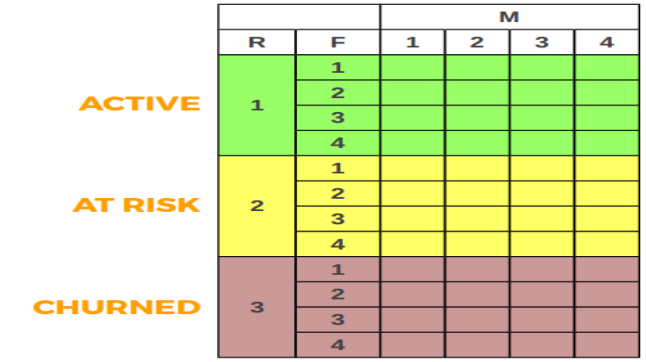
Step 3: Add segment numbers to the RFM Table

Customer	Recency	Frequency	Monetary	R	F	M
A	53 days	3 tran.	$230	2	2	2
B	120 days	10 tran.	$900	3	3	2
C	10 days	1 tran.	$20	1	1	1

This is called a **Segmented RFM Table**

Use the *Recency* segmentation to identify customers at risk of churn.

This works especially well if you use *Survival Analysis* for *Recency* segmentation.

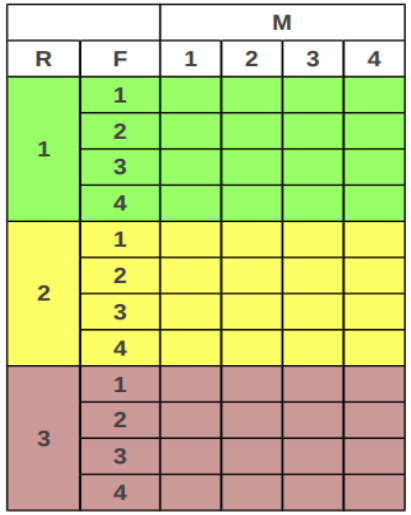


Also, use *Recency* for campaigning

X/UP-SELL, PROMOTIONAL ← ACTIVE

RETENTION CAMPAIGN ← AT RISK

REACTIVATION CAMPAIGN ← CHURNED



Use the *Frequency & Monetary* segmentation to estimate customer value.

Typical segment names: Premium, Gold, Silver etc.

		M			
R	F	1	2	3	4
1	1			SILVER	SILVER
	2		SILVER	SILVER	GOLD
	3	SILVER	SILVER	GOLD	GOLD
	4	SILVER	GOLD	GOLD	PREMIUM

RFM-SEGMENTATION STEPS

Each transaction will move customers through Recency and Value tiers.

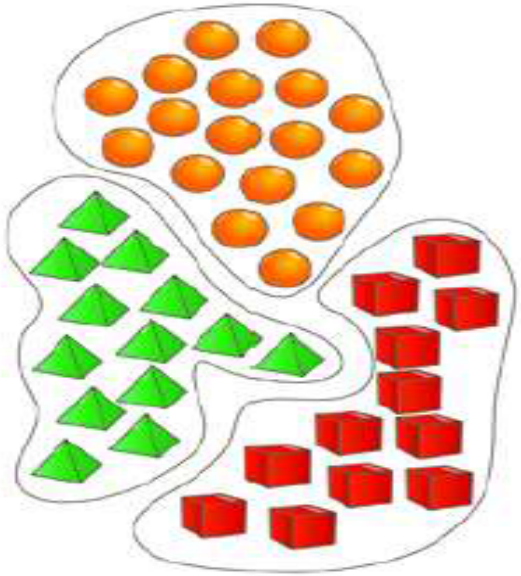
With RFM Metrics based on sums of events, the move can only be towards higher valued segments.

Behavioral Segmentation - Clustering Techniques

- K-means
  - Iteratively re-assign points to the nearest cluster center
- Agglomerative clustering(Heirarchical)
  - Start with each point as its own cluster and iteratively merge the closest clusters
- Mean-shift clustering
  - Estimate modes of pdf
- Spectral clustering
  - Split the nodes in a graph based on assigned links with similarity weights

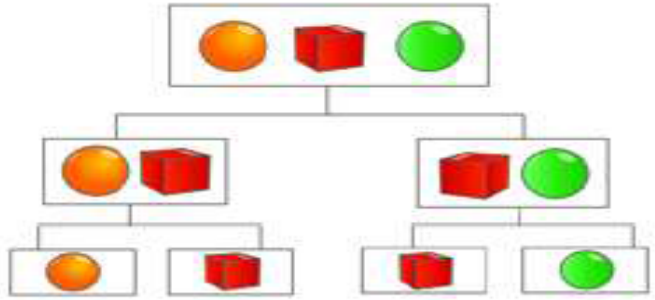
As we go down this chart, the clustering strategies have more tendency to transitively group points even if they are not nearby in feature space

Behavioral Segmentation: Hierarchical Vs. Non-hierarchical



- ***Partitional* clustering or non-hierarchical :** A division of objects into non-overlapping subsets (clusters) such that each object is in exactly one cluster
- The non-hierarchical methods divide a dataset of N objects into M clusters.
- **K-means clustering**, a non-hierarchical technique, is the most commonly used one in business analytics

- ***Hierarchical* clustering:** A set of nested clusters organized as a hierarchical tree
- The hierarchical methods produce a set of nested clusters in which each pair of objects or clusters is progressively nested in a larger cluster until only one cluster remains
- **CHAID tree** is most widely used in business analytics



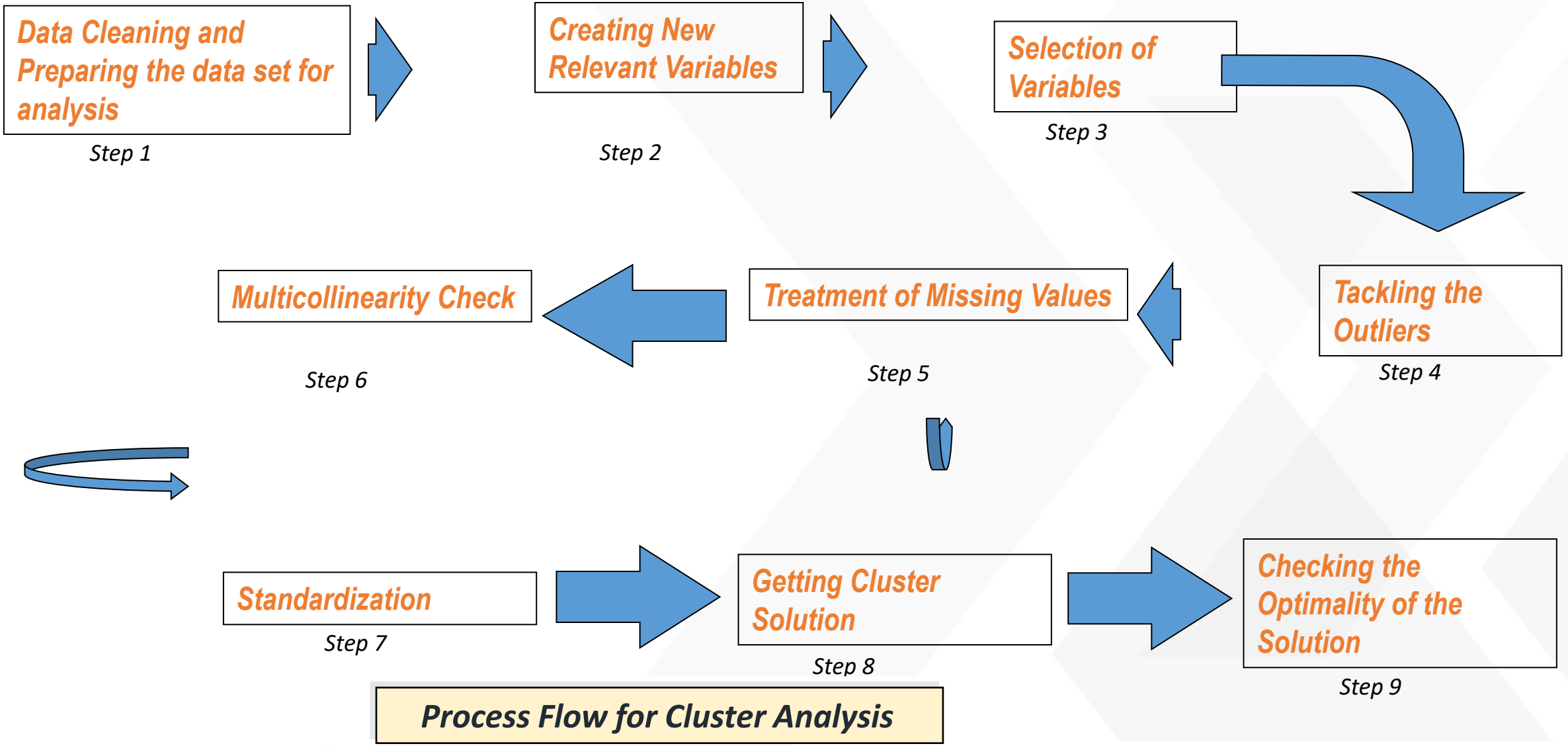
Behavioral Segmentation: Subjective Segmentation-Cluster Analysis

Highest value segment

	Big Ticket	Frequent	Small Ticket	Infrequent	Returner	Overall
% Customers	9.8	4.2	13.5	69.5	6.6	100.0
% Revenue	27.4	33.6	15.4	13.5	10.1	100.0
Revenue per customer ($)	1,038	8,618	1209.1	220	1613.5	1077.2
Visits per customer	3.1	34.2	16.1	2.1	8.3	4.8
Basket size ($)	970.1	252.7	75.1	105.2	165.1	224.8
Average departments shopped	3.6	5.5	1.9	1.2	2.9	1.9
Stores shopped	1.1	3.0	1.8	1.1	1.2	1.7
Returning propensity (%)	0.3	6.5	5.5	0.3	25.5	3.2
Shopped in December (%)	15.1	70.8	53.3	19.4	23.3	26.6
Shopped on Memorial Day (%)	1.6	17.9	2.4	0.9	2.1	2.2
Shopped on Labor Day (%)	1.0	14.1	1.8	0.6	1.5	1.7
Shopped on President's Day (%)	0.7	12.0	1.8	0.6	1.8	1.5
Average Discount Rate (%)	14.8	11.4	6.6	4.5	10.6	11.2
Customer lifetime (months)	25.2	46.2	42.2	28.4	27.2	30.8

Note that key *profile* variables are not always the same as *basis* variables used to generate the segmentation

Subjective Segmentation: Cluster Analysis Process



Process Flow for Cluster Analysis

Subjective Segmentation: K-Means Clustering Algorithm

1. The number *k* of clusters is fixed
2. An initial set of *k "seeds" (aggregation centres)* is provided
    1. First *k* elements
    2. Other seeds (randomly selected or explicitly defined)
3. Given a certain fixed threshold, all units are assigned to the nearest cluster seed
4. New seeds are computed
5. Go back to step 3 until no reclassification is necessary

Or simply

Initialize k cluster centers

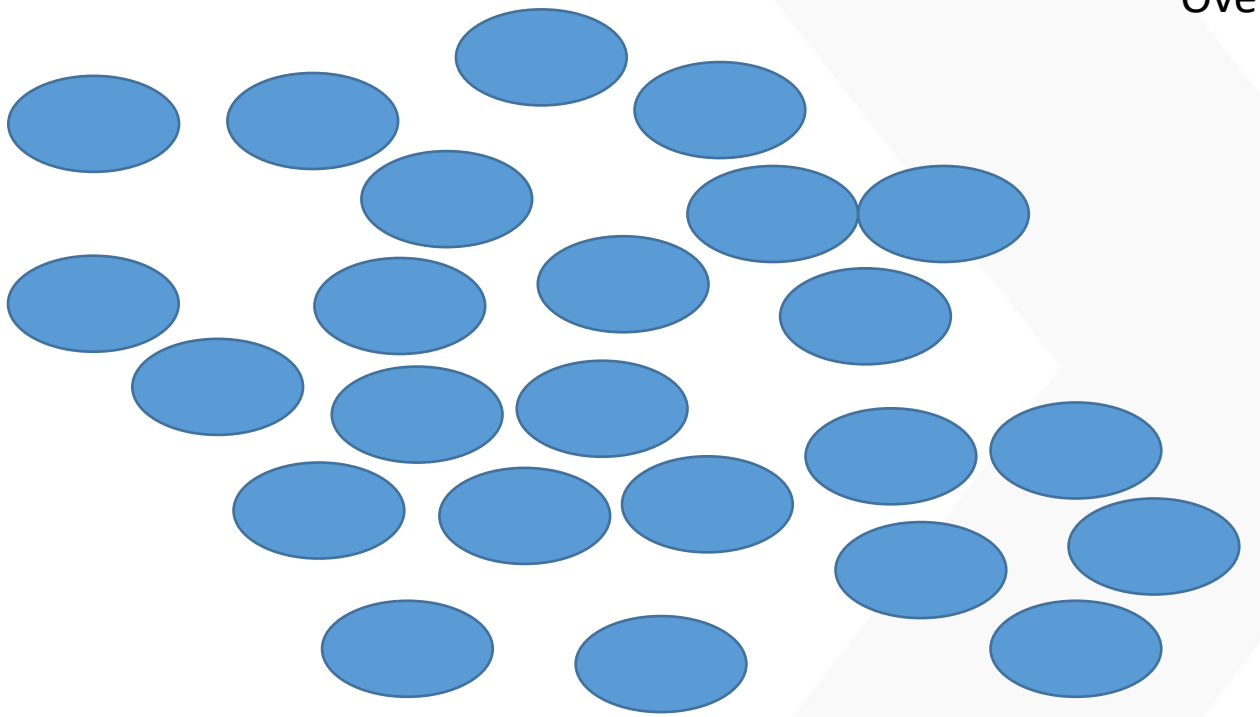
Do

Assignment step: Assign each data point to its closest cluster center

Re-estimation step: Re-compute cluster centers

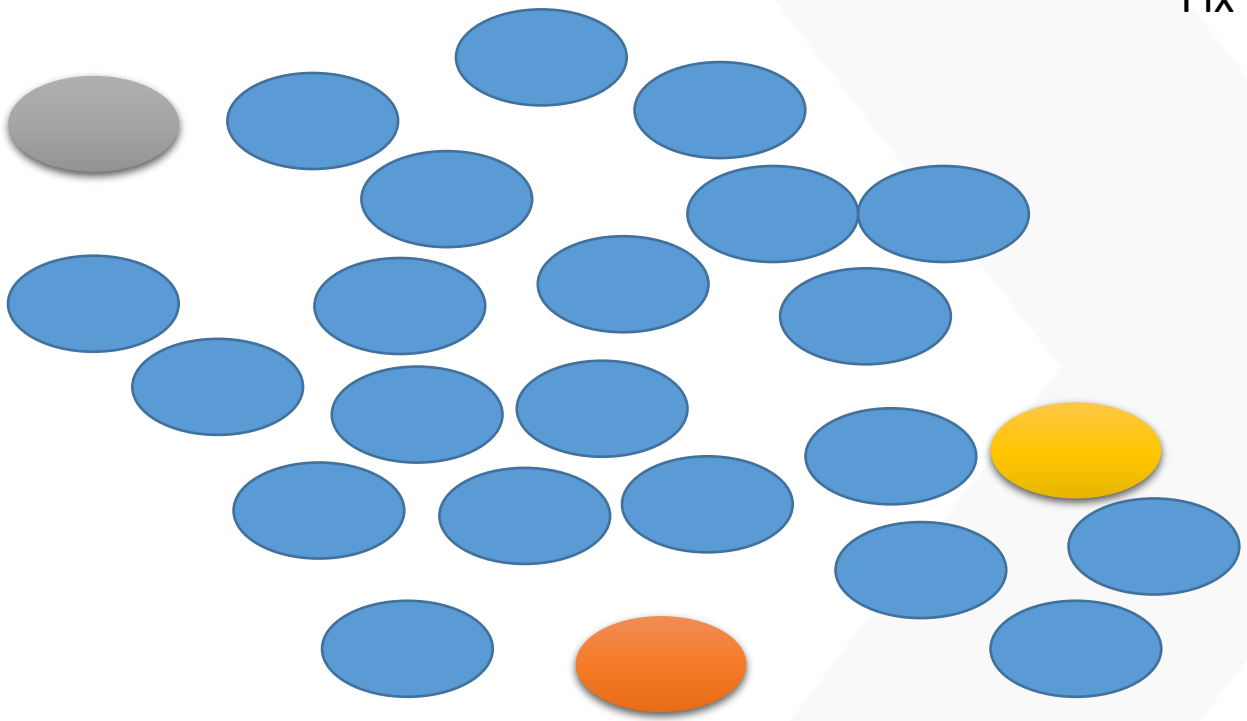
While (there are still changes in the cluster centers)

K-Means clustering



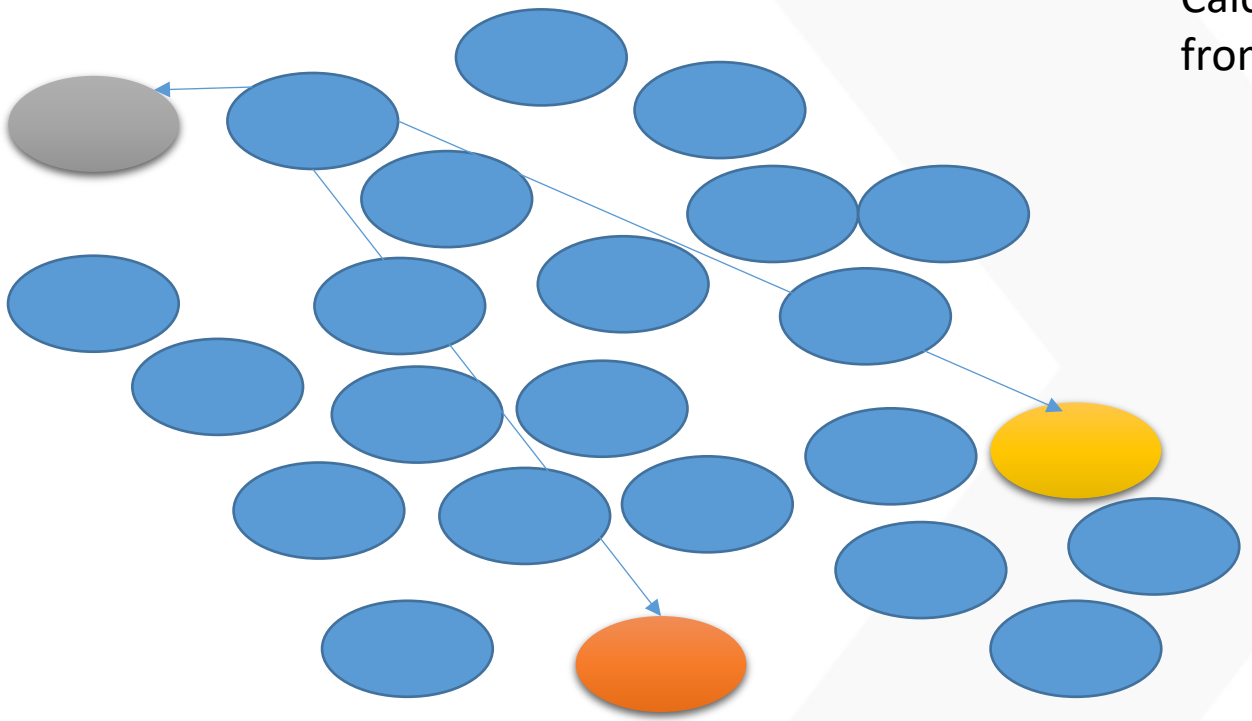
Overall population

K-Means clustering



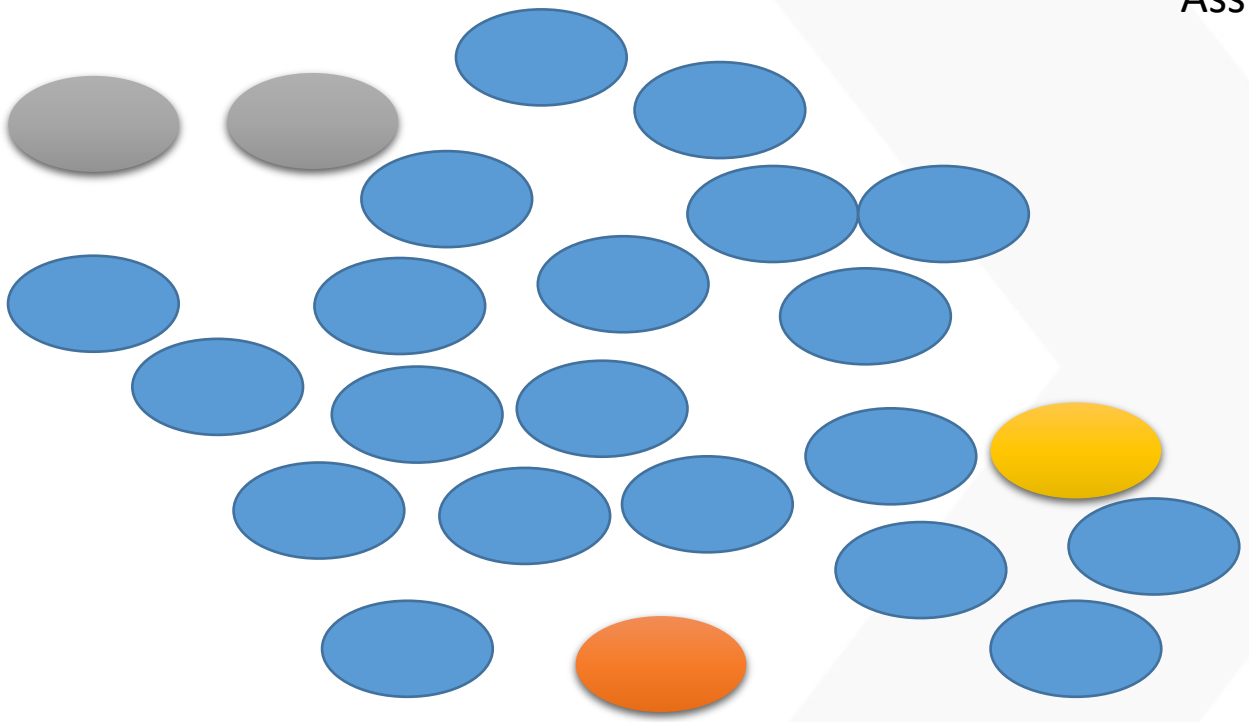
Fix the number of clusters

K-Means clustering



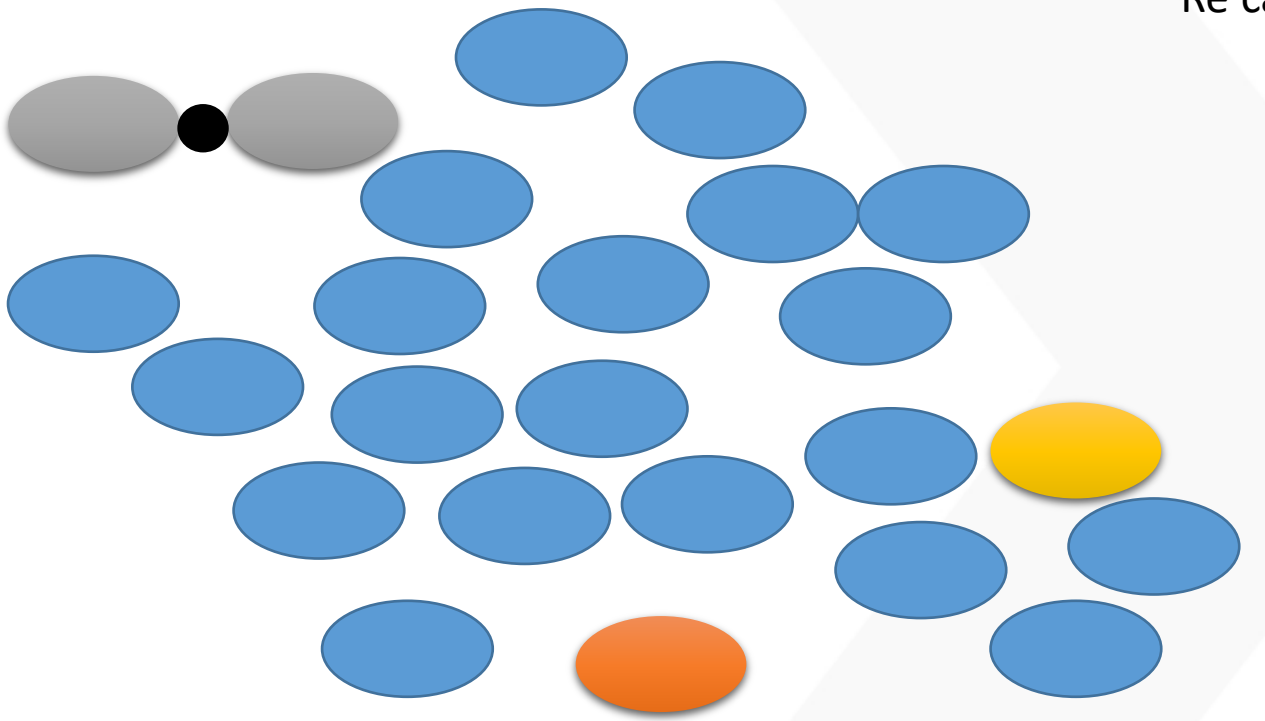
Calculate the distance of each case from all clusters

K-Means clustering



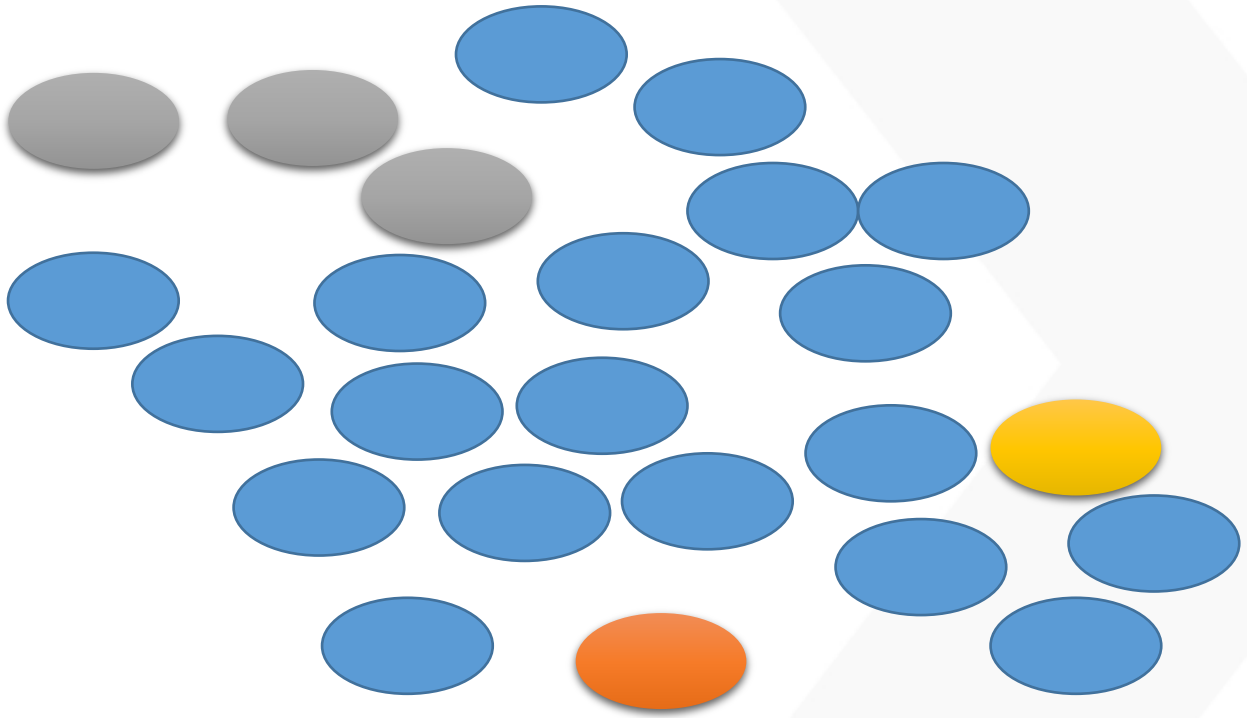
Assign each case to nearest cluster

K-Means clustering

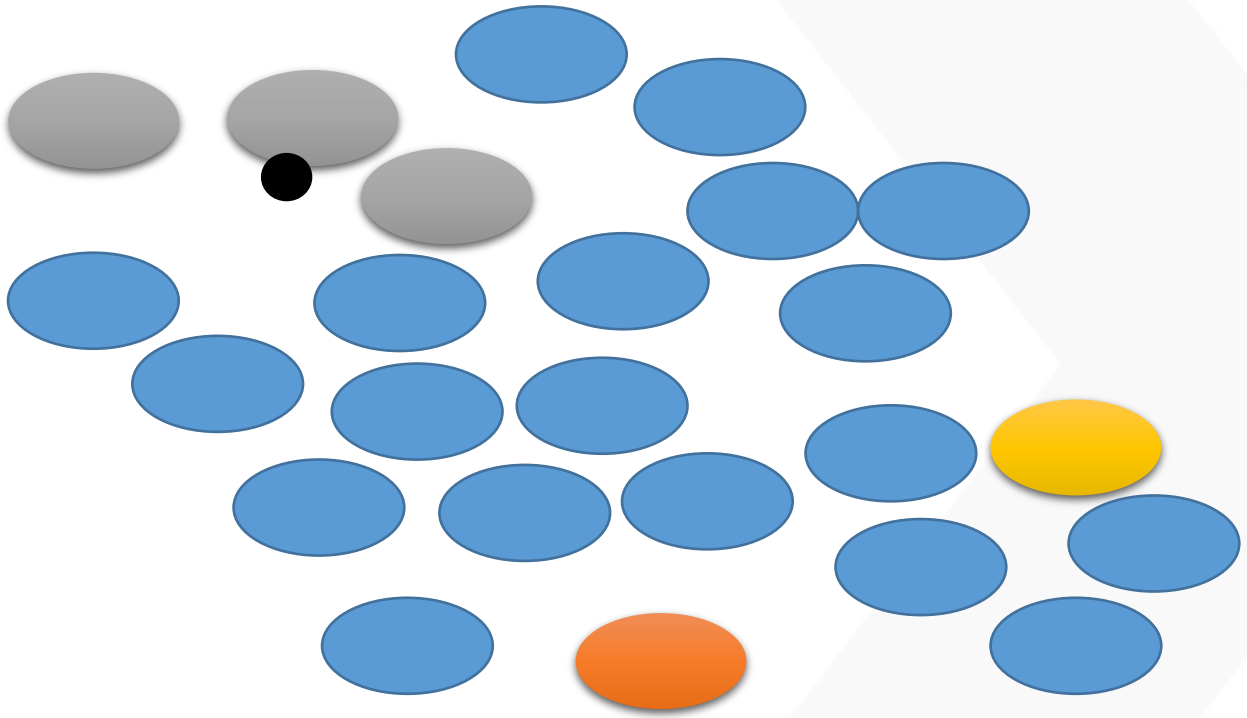


Re calculate the cluster centers

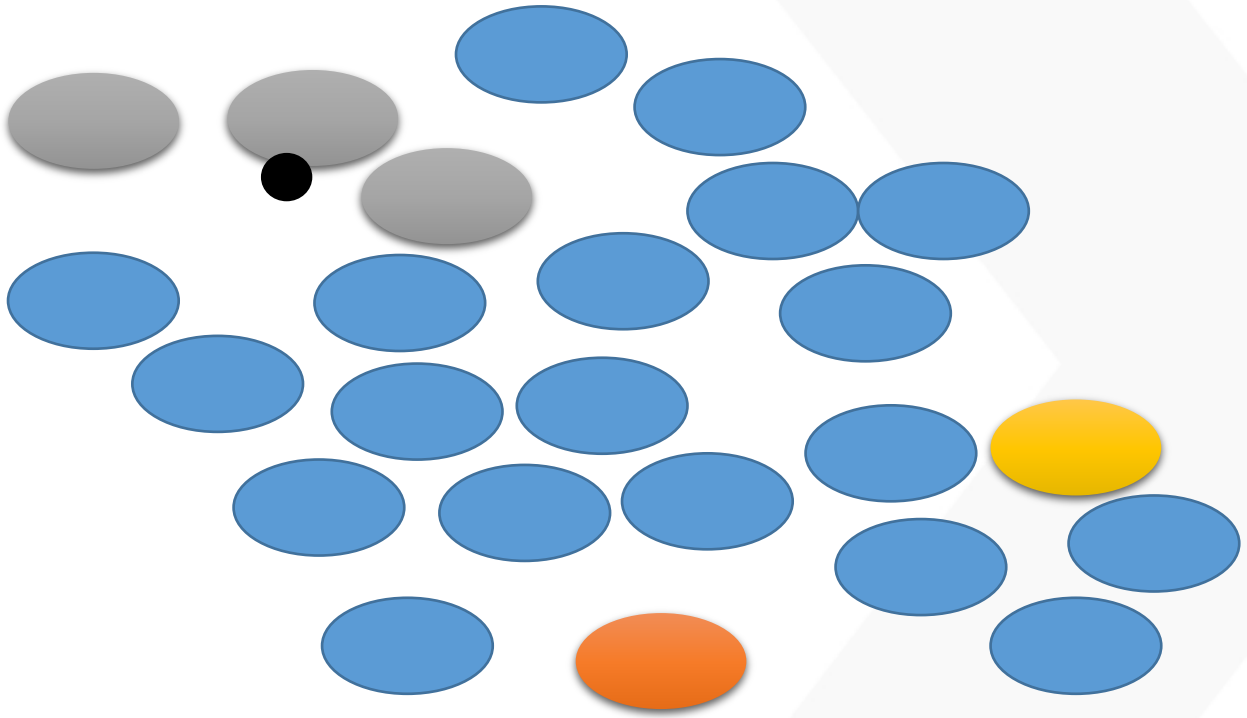
K-Means clustering



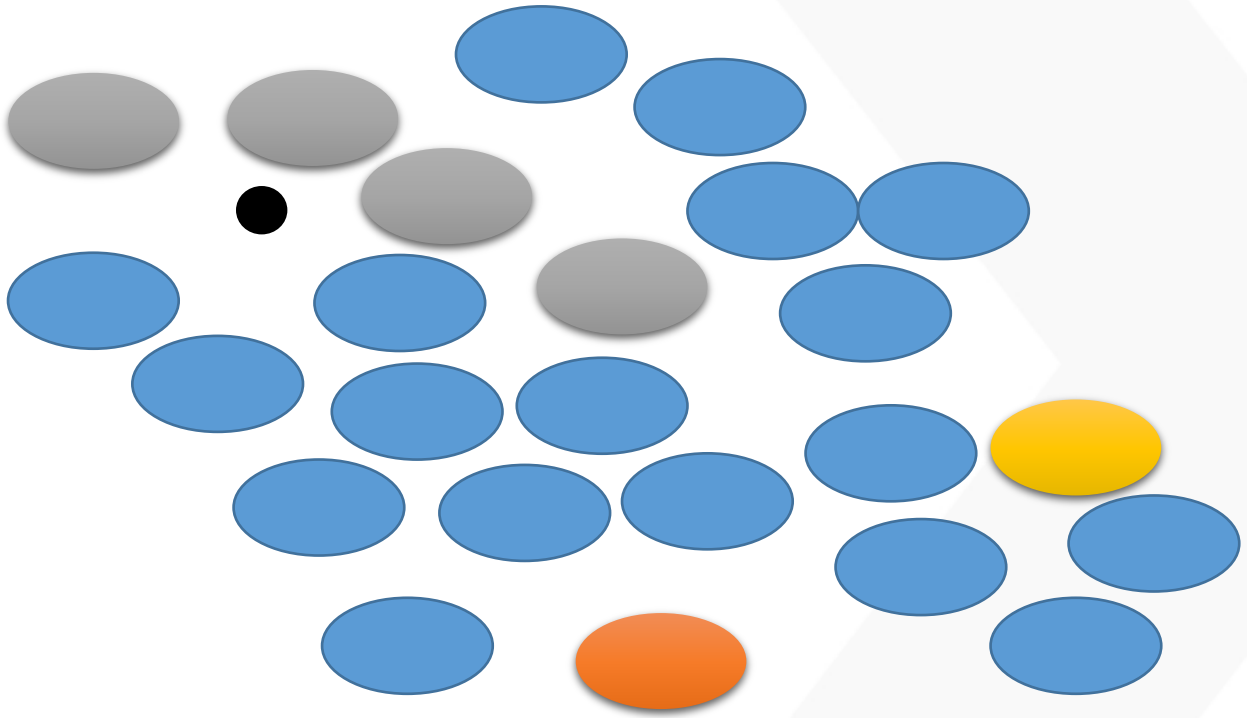
K-Means clustering



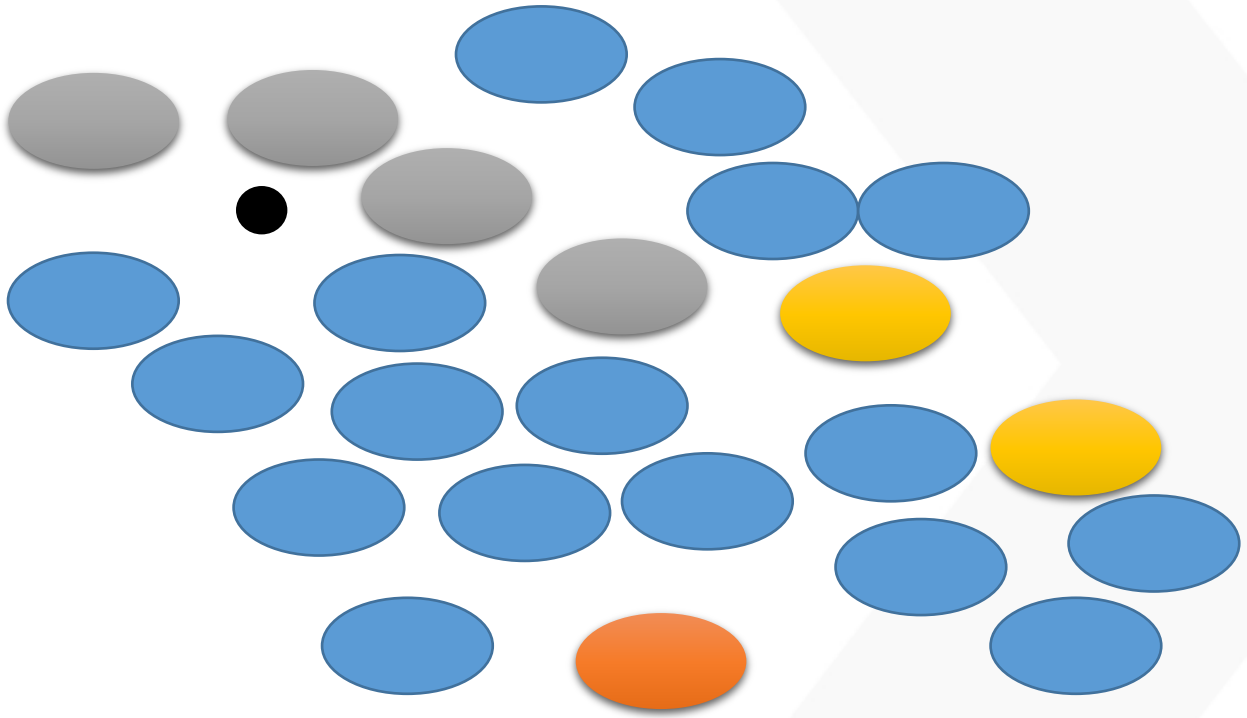
K-Means clustering



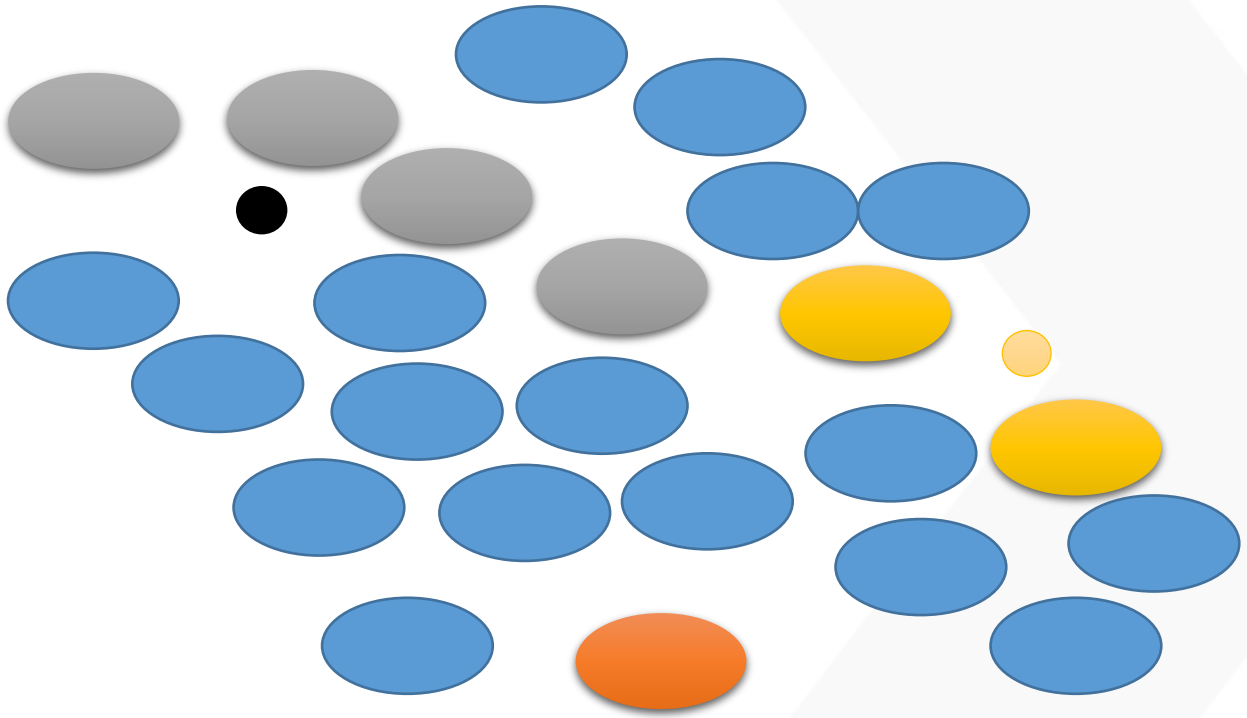
K-Means clustering



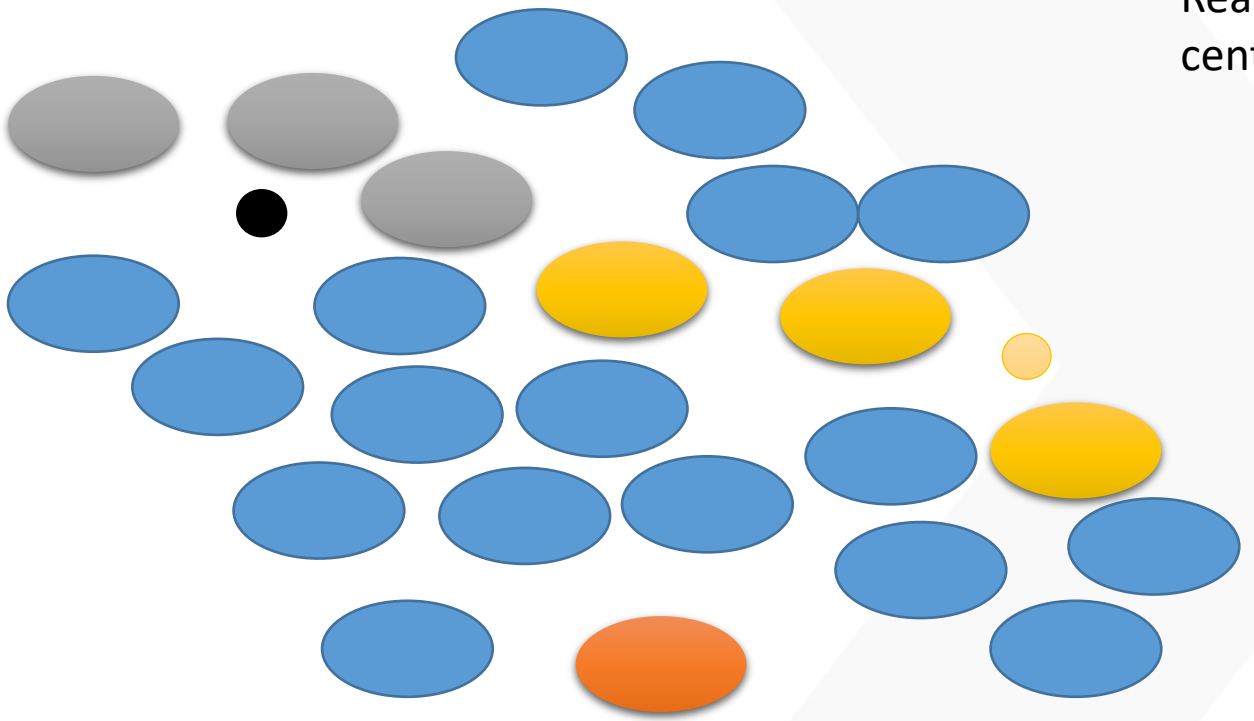
K-Means clustering



K-Means clustering

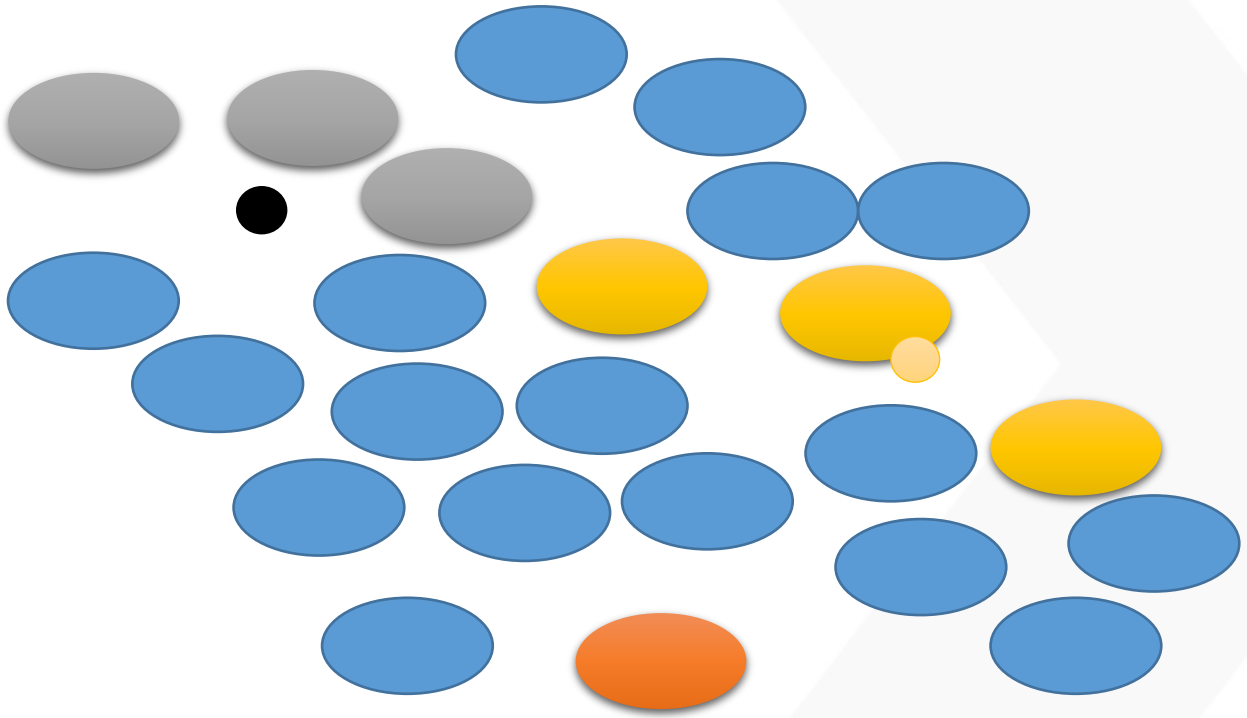


K-Means clustering

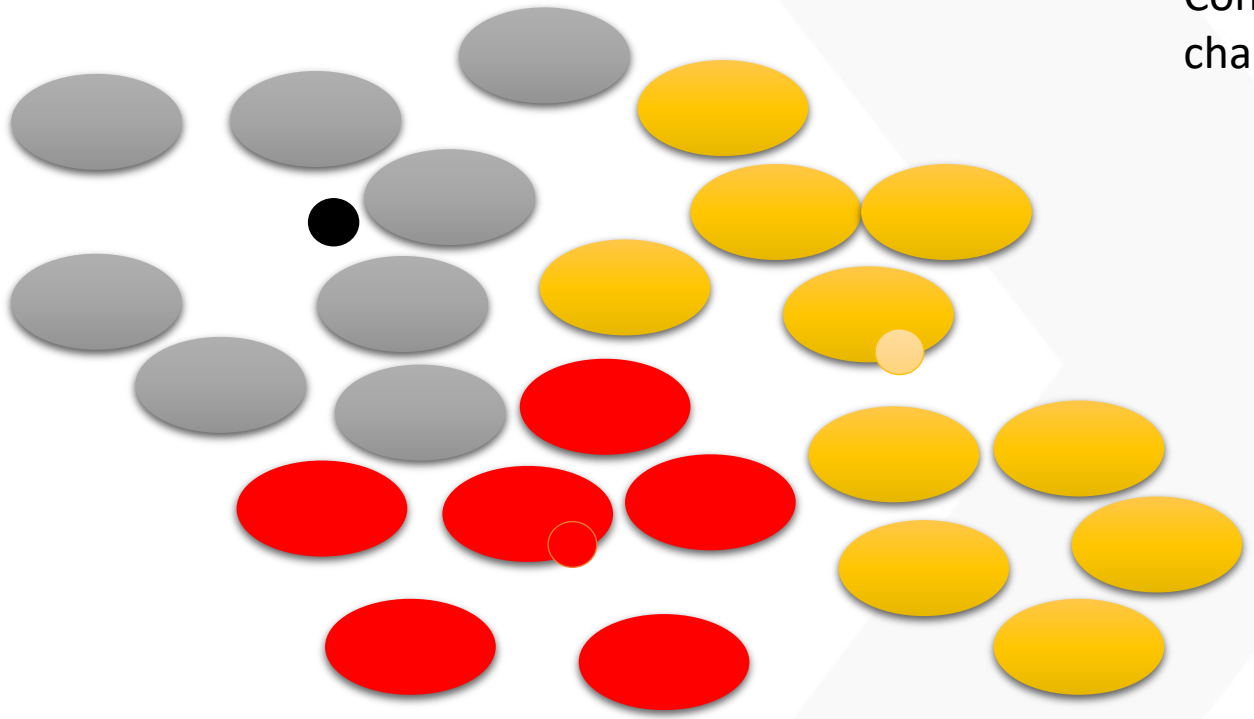


Reassign after changing the cluster centers

K-Means clustering



K-Means clustering



Continue till there is no significant change between two iterations

Calculating the distance

	Weight
Cust1	68
Cust2	72
Cust3	100

Which of the two customers are similar?

	Weight	Age
Cust1	68	25
Cust2	72	70
Cust3	100	28

Which of the two customers are similar now?

	Weight	Age	Income
Cust1	68	25	60,000
Cust2	72	70	9,000
Cust3	100	28	62,000

Which two of the customers are similar in this case?

Distance Measures

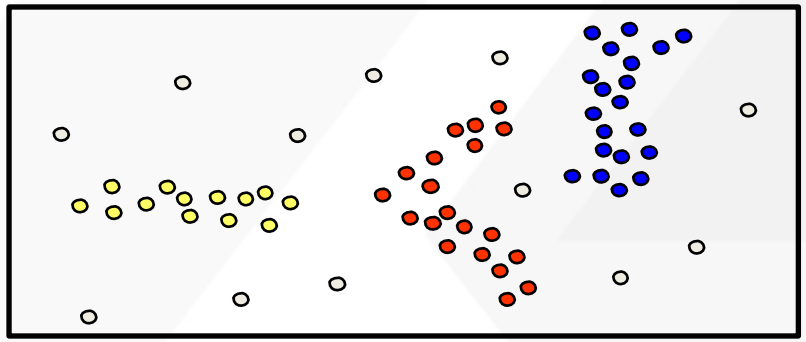
- Euclidean distance
- City-block (Manhattan) distance
- Chebychev similarity
- Minkowski distance
- Mahalanobis distance
- Maximum distance
- Cosine similarity
- Simple correlation between observations
- Minimum distance
- Weighted distance

Not sure all these measures will result in same clusters in the above example

Spectral Clustering (Density Based Clustering)

Density-based Clustering

- **Basic idea**
  - Clusters are dense regions in the data space, separated by regions of lower object density
  - A cluster is defined as a maximal set of density- connected points
  - Discovers clusters of arbitrary shape
- **Method**
  - DBSCAN



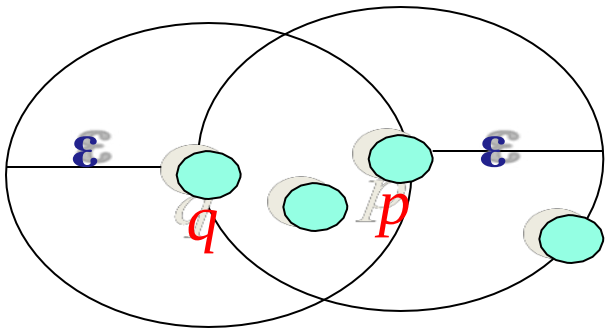
Density Definition

- ε-Neighborhood – Objects within a radius of ε from an object.

$$N_\varepsilon(p) : \{q \mid d(p,q) \leq \varepsilon\}$$

- High density - ε-Neighborhood of an object contains at least *MinPts* of objects.

MinPts



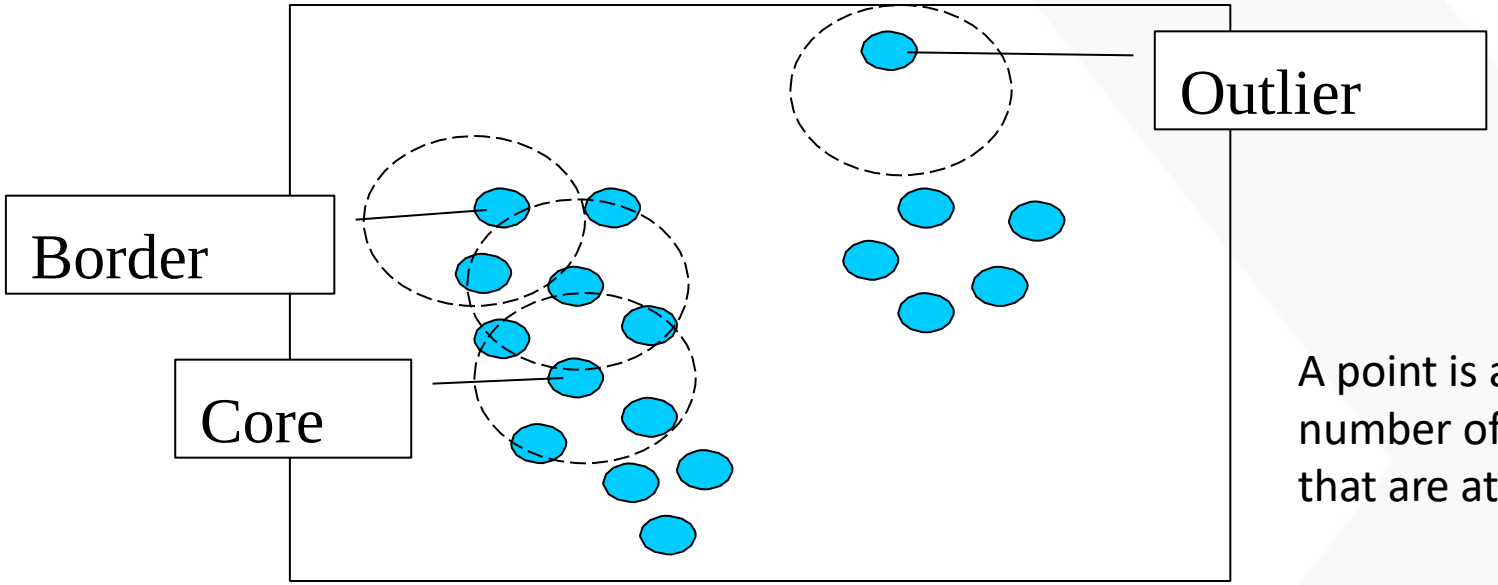
ε-Neighborhood of *p*

ε-Neighborhood of *q*

Density of p is "high" (MinPts = 4)

Density of q is "low" (MinPts = 4)

Core, Border & Outlier



Given ε and *MinPts*, categorize the objects into three exclusive groups.

A point is a core point if it has more than a specified number of points (MinPts) within Eps—These are points that are at the interior of a cluster.

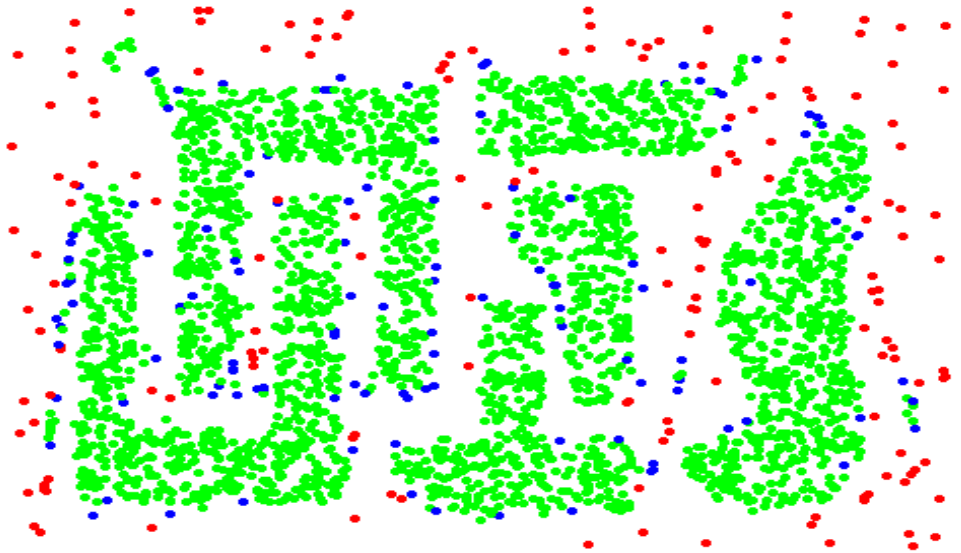
$\varepsilon = 1$unit, MinPts = 5

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.

A noise point is any point that is not a core point nor a border point.

Example

Original Points

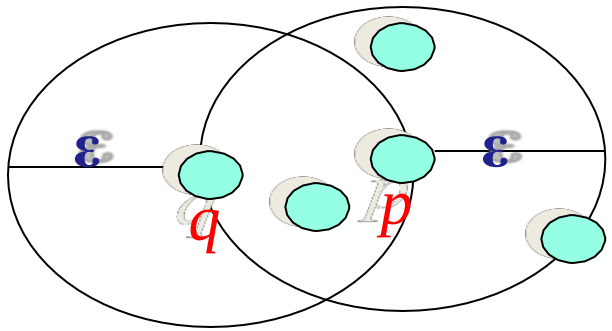


Point types: core, border and outliers

ε = 10, MinPts = 4

Density-reachability

- Directly density-reachable
  - An object *q* is directly density-reachable from object *p* if *p* is a core object and *q* is in p's ε-neighborhood.

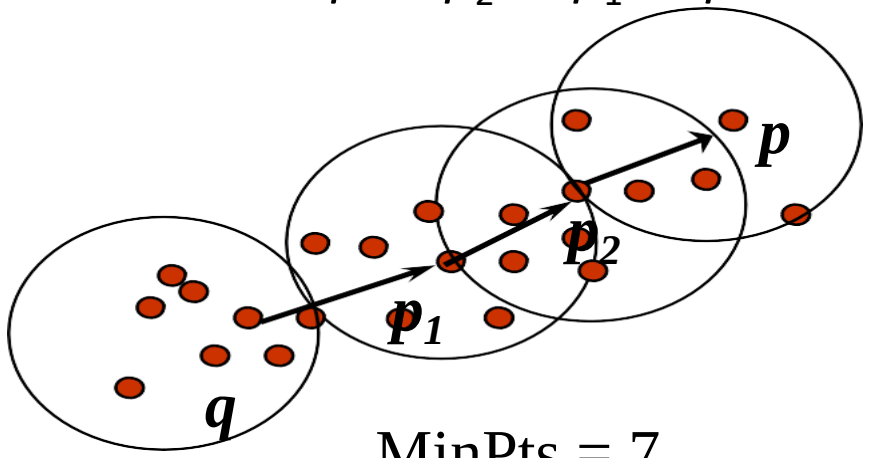


MinPts = 4

- *q* is directly density-reachable from *p*
- *p* is not directly density-reachable from *q*
- Density-reachability is asymmetric

Density-reachability

- Density-Reachable (directly and indirectly):
  - A point p is directly density-reachable from p_2
  - p_2 is directly density-reachable from p_1
  - p_1 is directly density-reachable from q
  - $p \leftarrow p_2 \leftarrow p_1 \leftarrow q$ form a chain

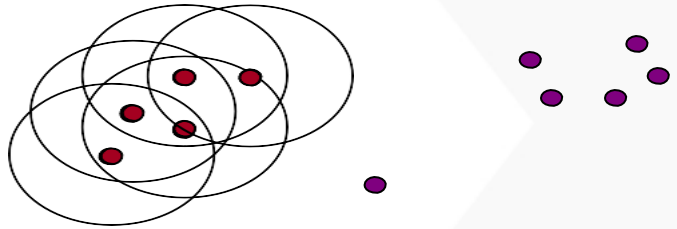


MinPts = 7

- p is (indirectly) density-reachable from q
- q is not density-reachable from p

DBSCAN Algorithm: Example

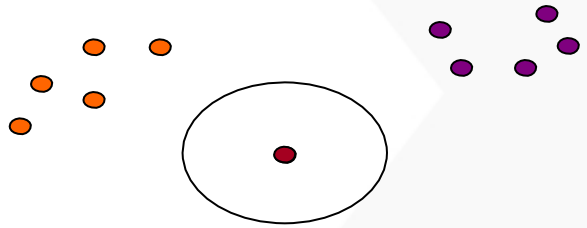
- **Parameter**
  - ε = 2 cm
  - *MinPts* = 3



```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
                else
            assign o to NOISE
```

DBSCAN Algorithm: Example

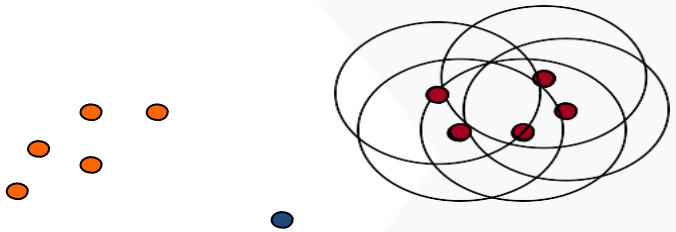
- **Parameter**
  - ε = 2 cm
  - *MinPts* = 3



for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
        else
      assign o to NOISE

DBSCAN Algorithm: Example

- **Parameter**
  - ε = 2 cm
  - *MinPts* = 3



for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
                else
            assign o to NOISE

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

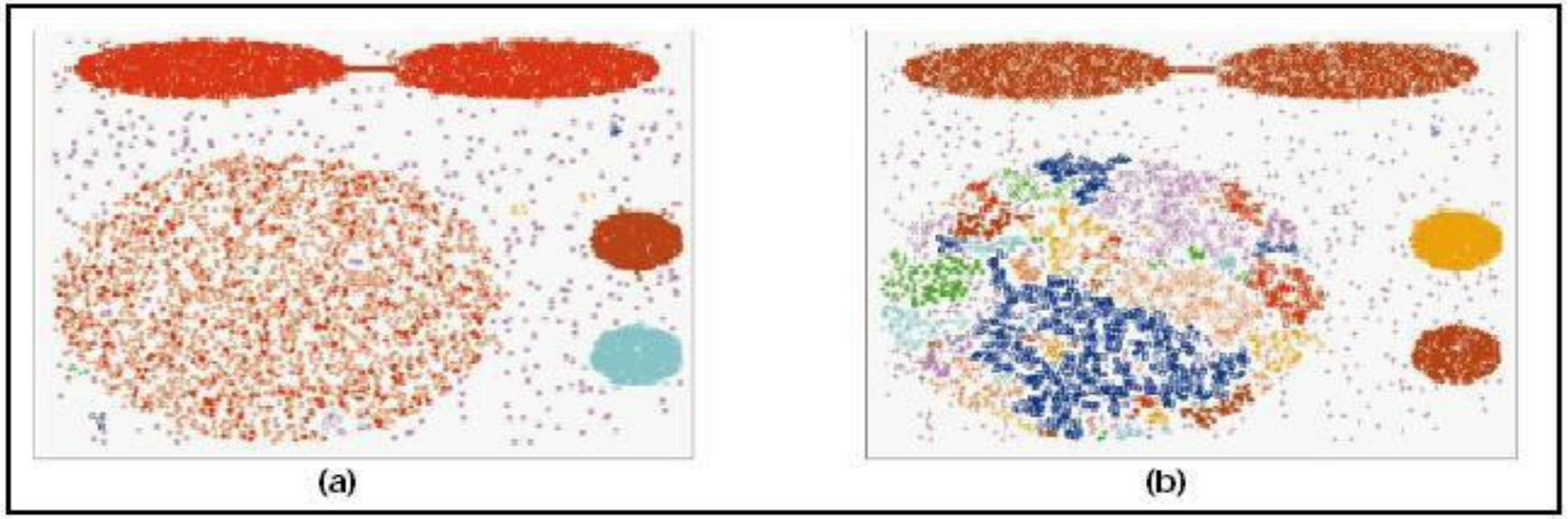
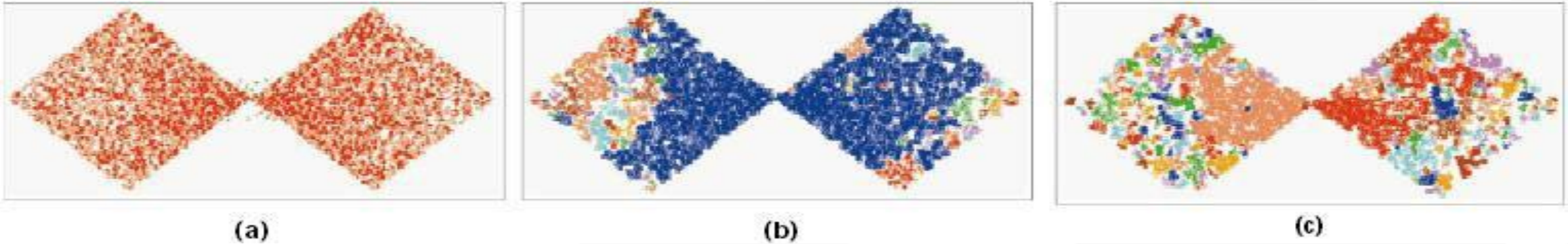
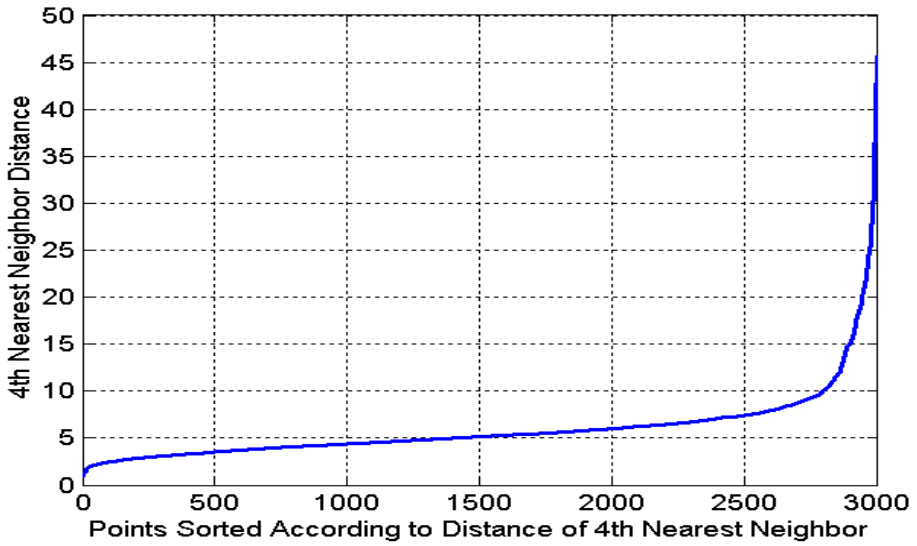


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



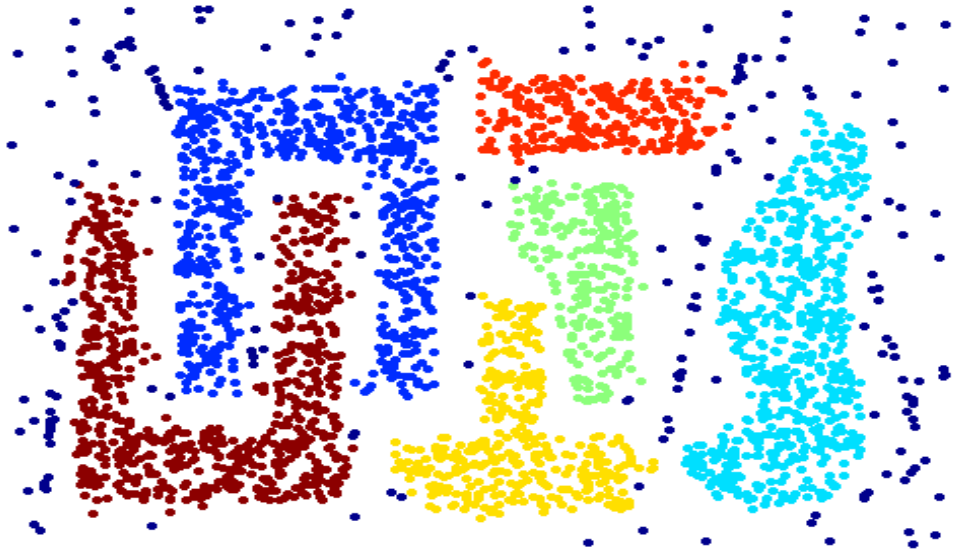
DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor



When DBSCAN Works Well

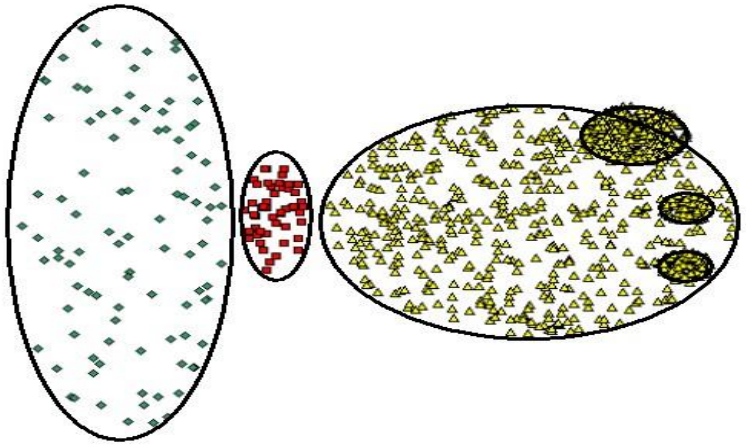
Original Points



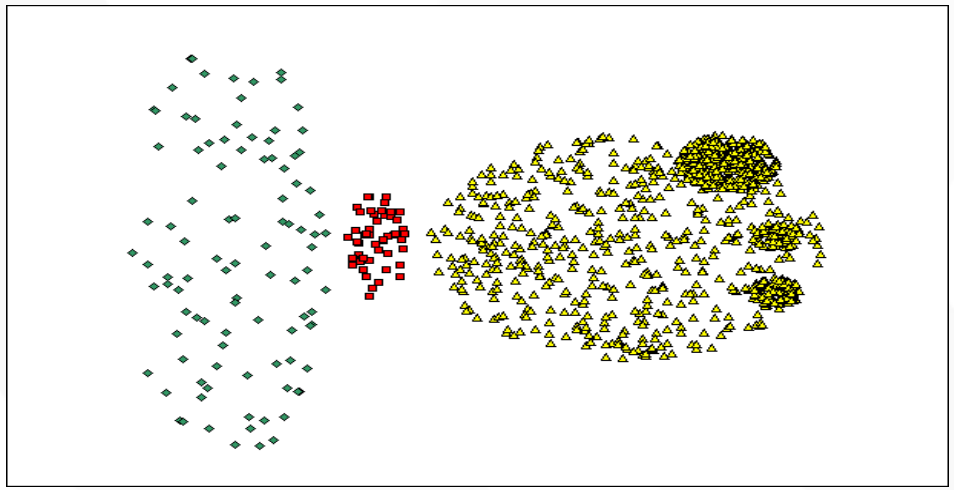
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

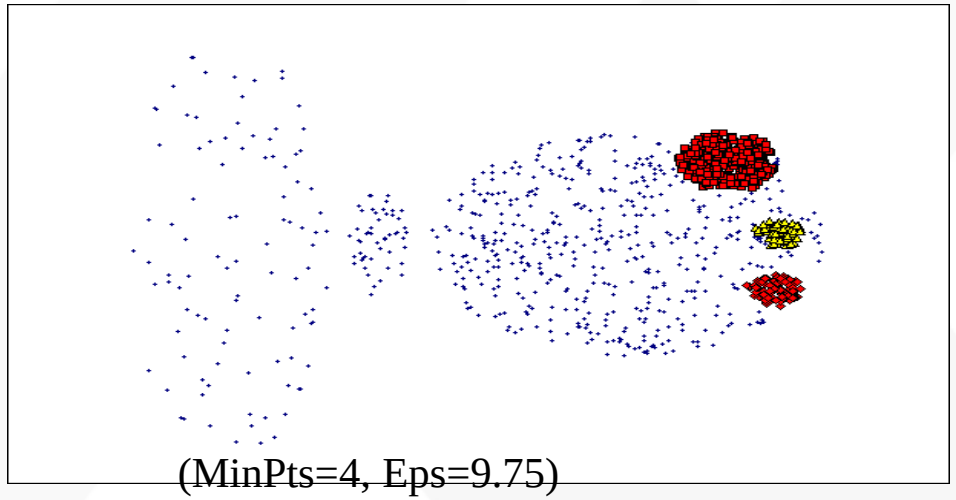
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

- Cannot handle varying densities
- sensitive to parameters—hard to determine the correct set of parameters

Take-away Message

- The basic idea of density-based clustering
- The two important parameters and the definitions of neighborhood and density in DBSCAN
- Core, border and outlier points
- DBSCAN algorithm
- DBSCAN’s pros and cons

Objective Segmentation

Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

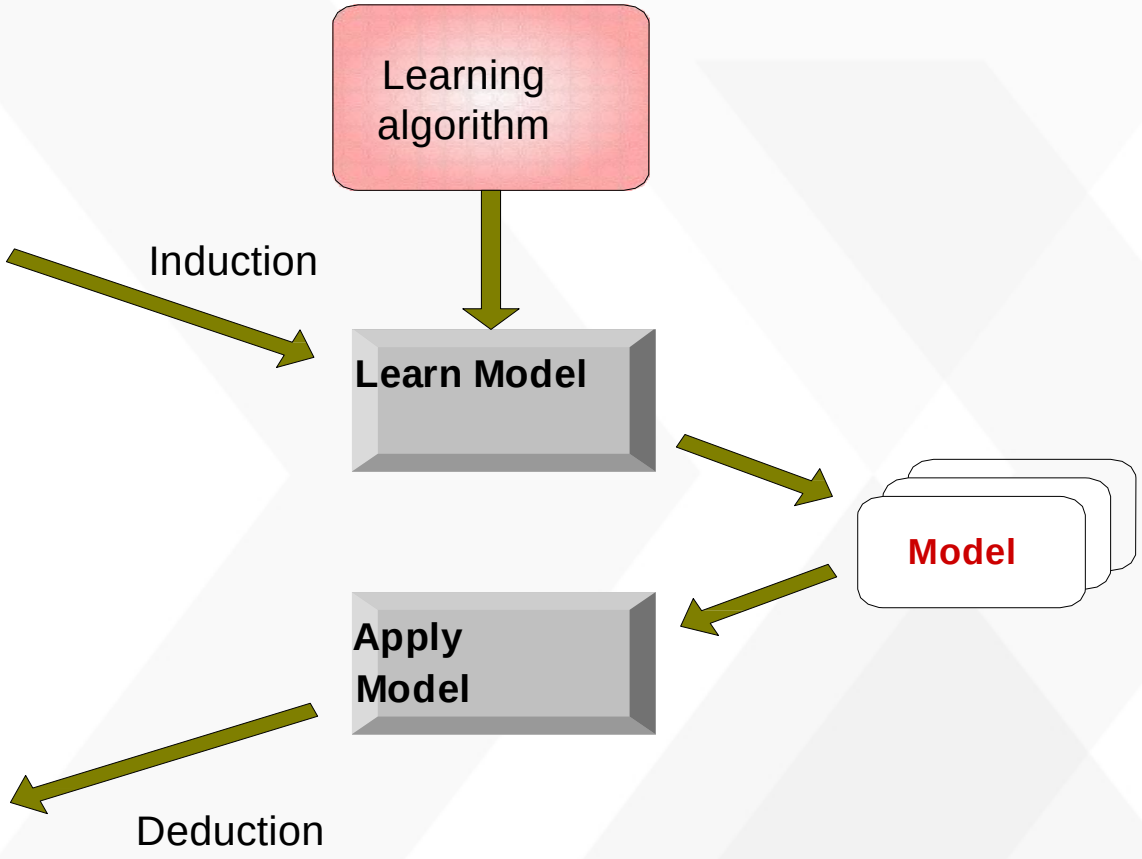
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Examples of Classification Task

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Classifying emails as spams or normal emails
- Categorizing news stories as finance, weather, entertainment, sports, etc

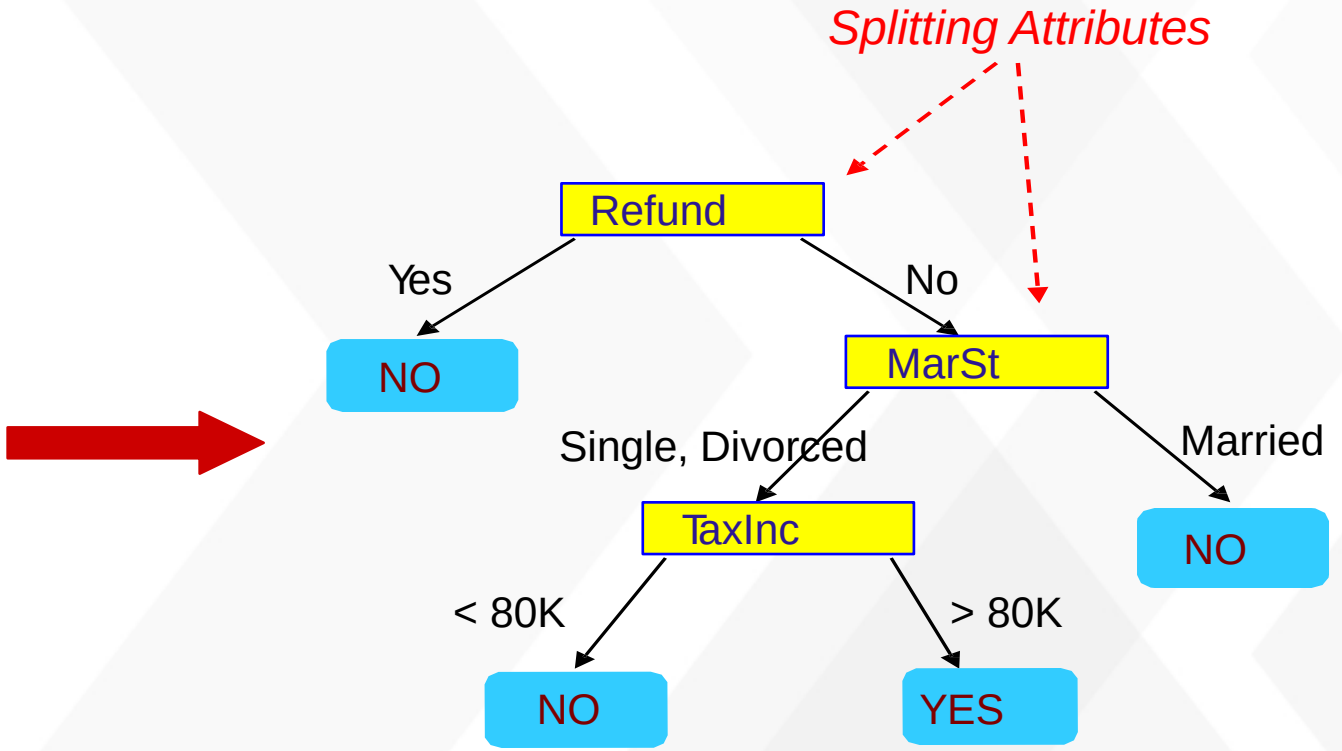
Classification Techniques

- Decision Tree
- Naïve Bayes
- Nearest Neighbor
- Rule-based Classification
- Logistic Regression
- Support Vector Machines
- Ensemble methods
- ……

Example of a Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

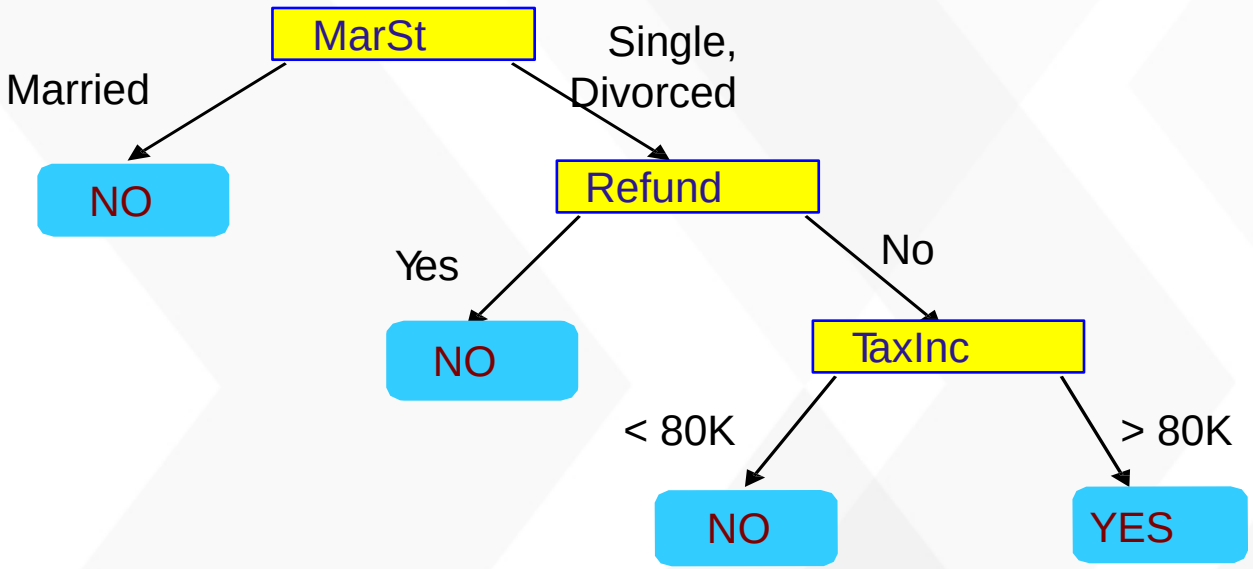
Training Data



Model: Decision Tree

Another Example of Decision Tree

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**



There could be more than one tree that fits the same data!

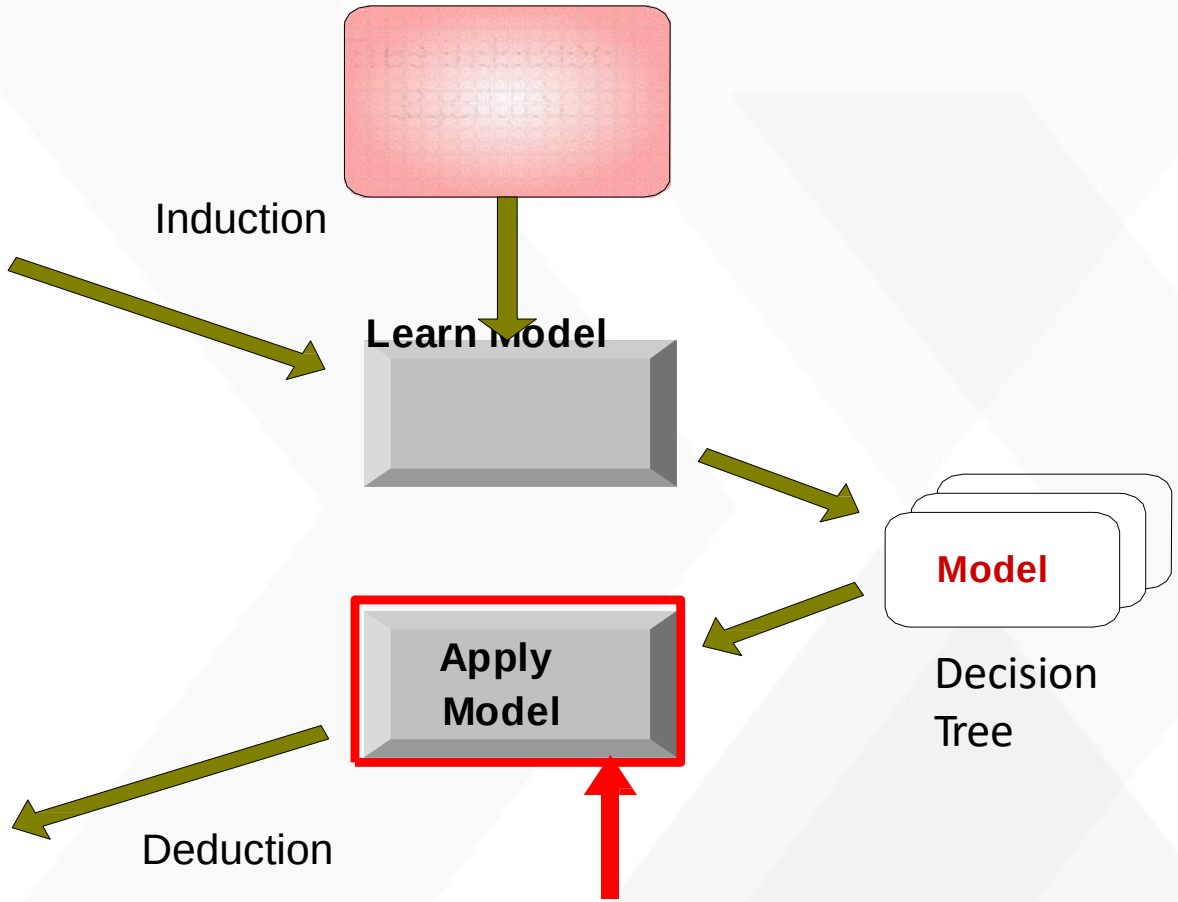
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

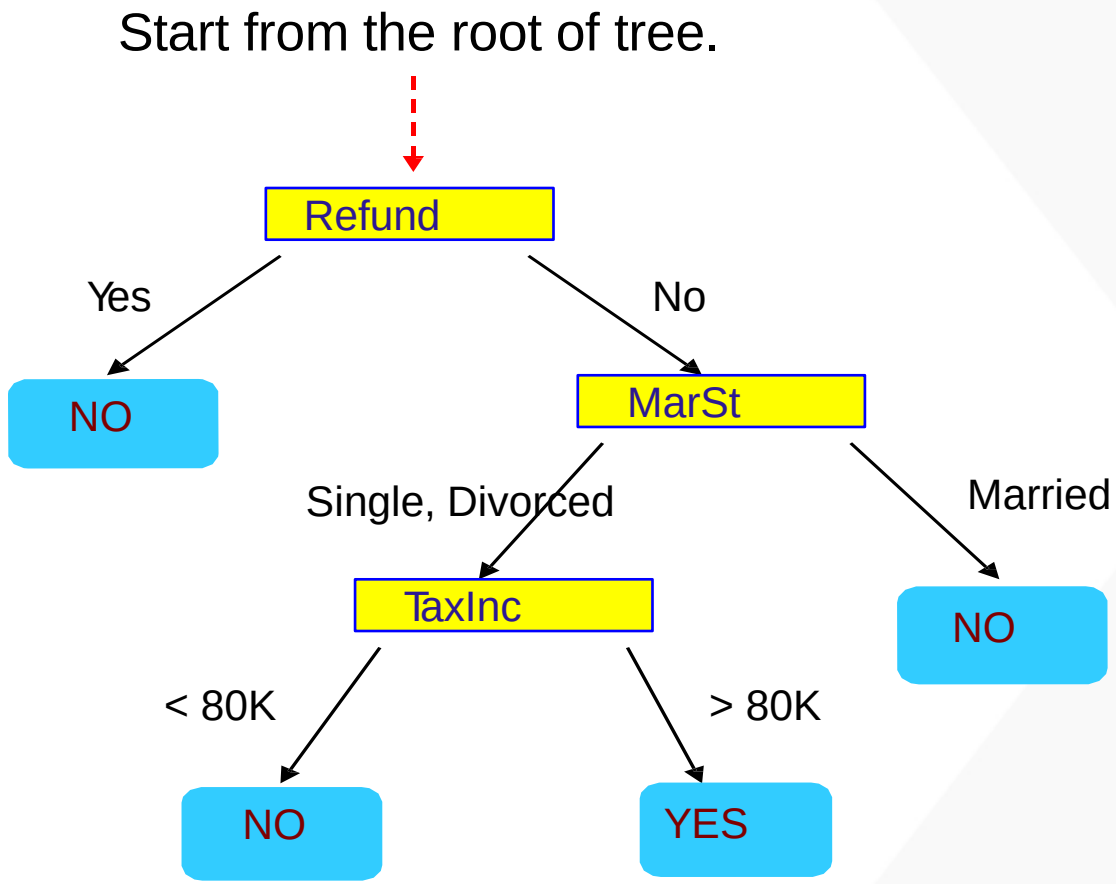
Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



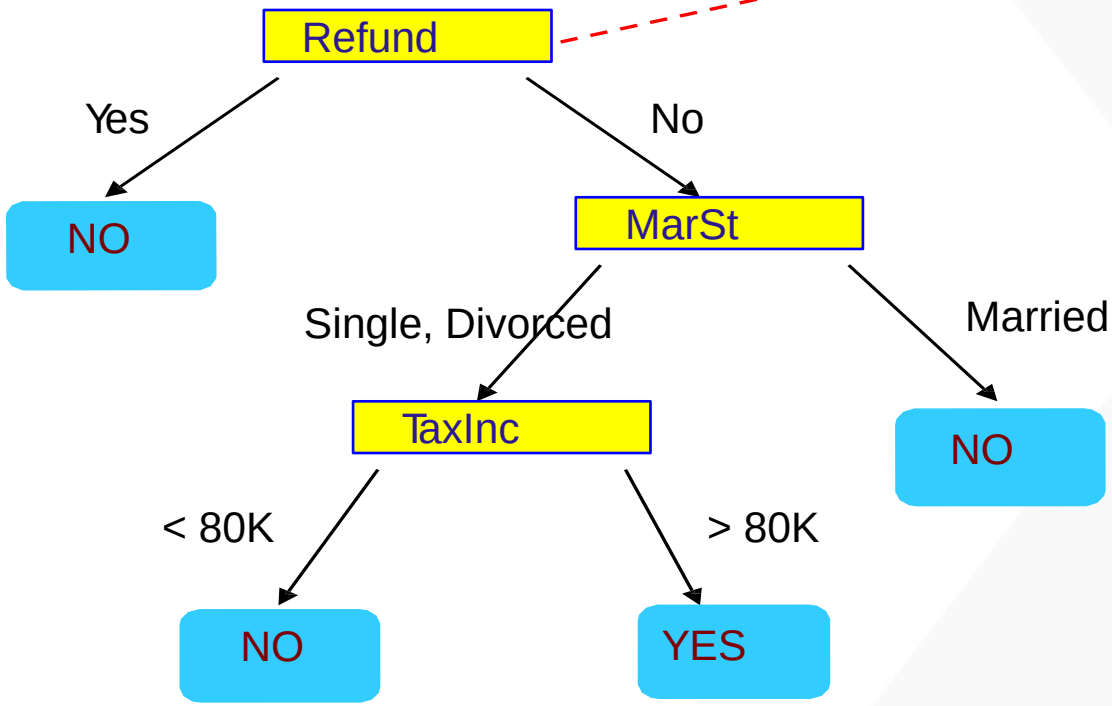
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

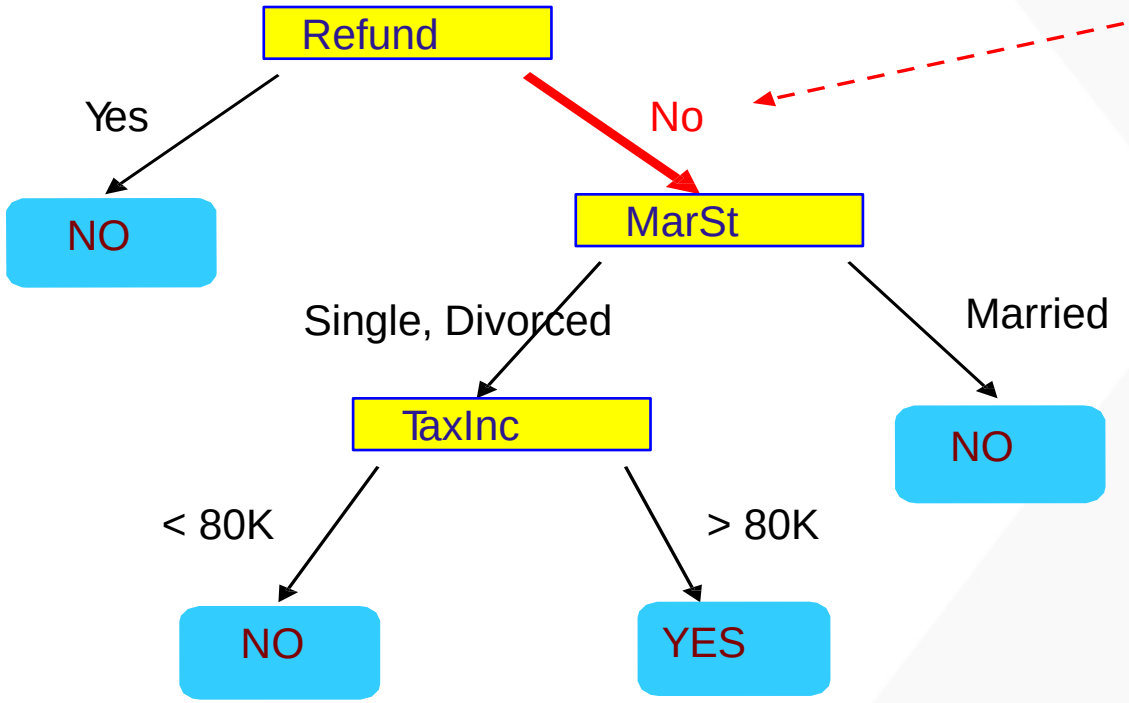
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

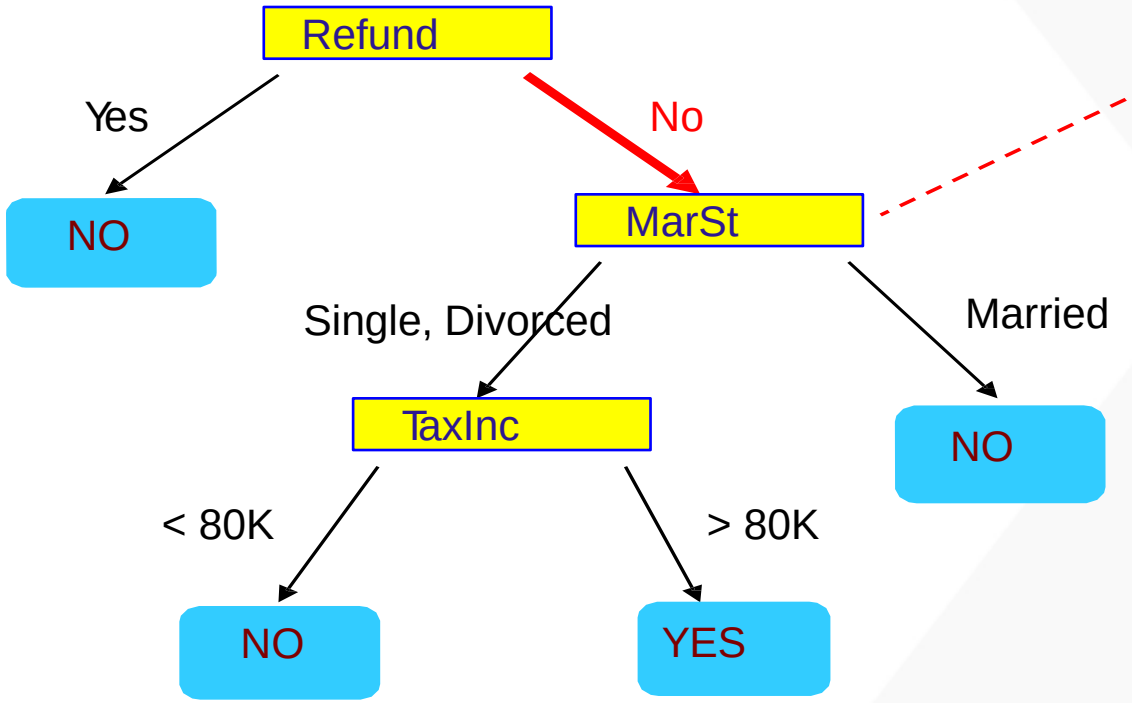
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

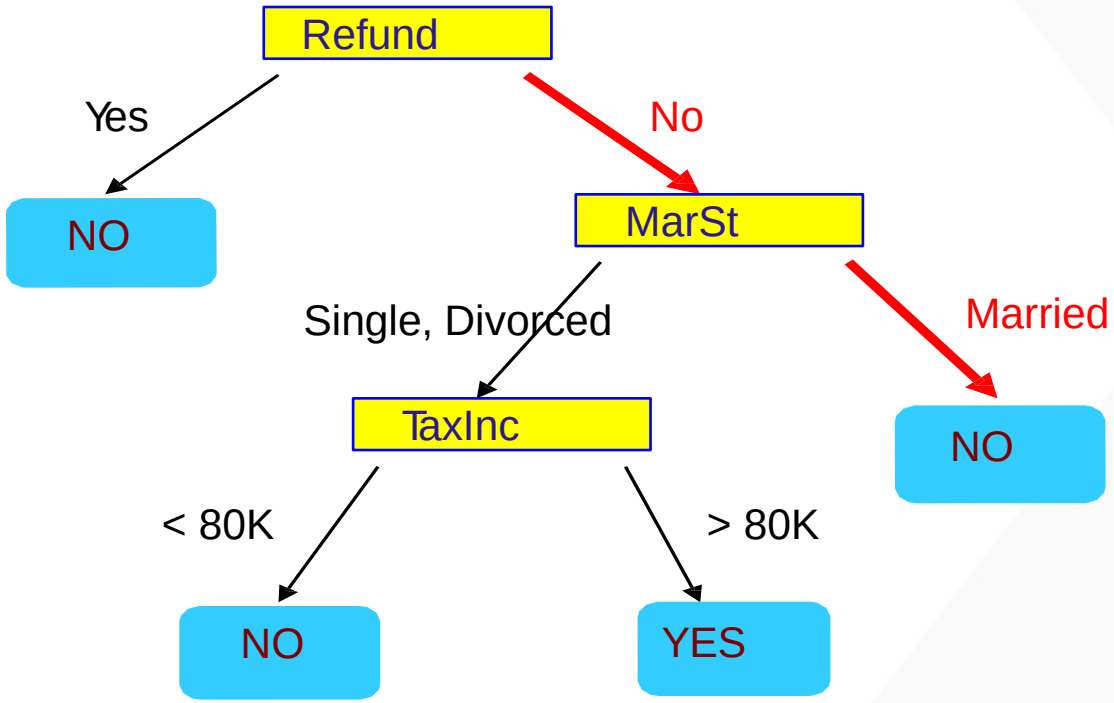
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

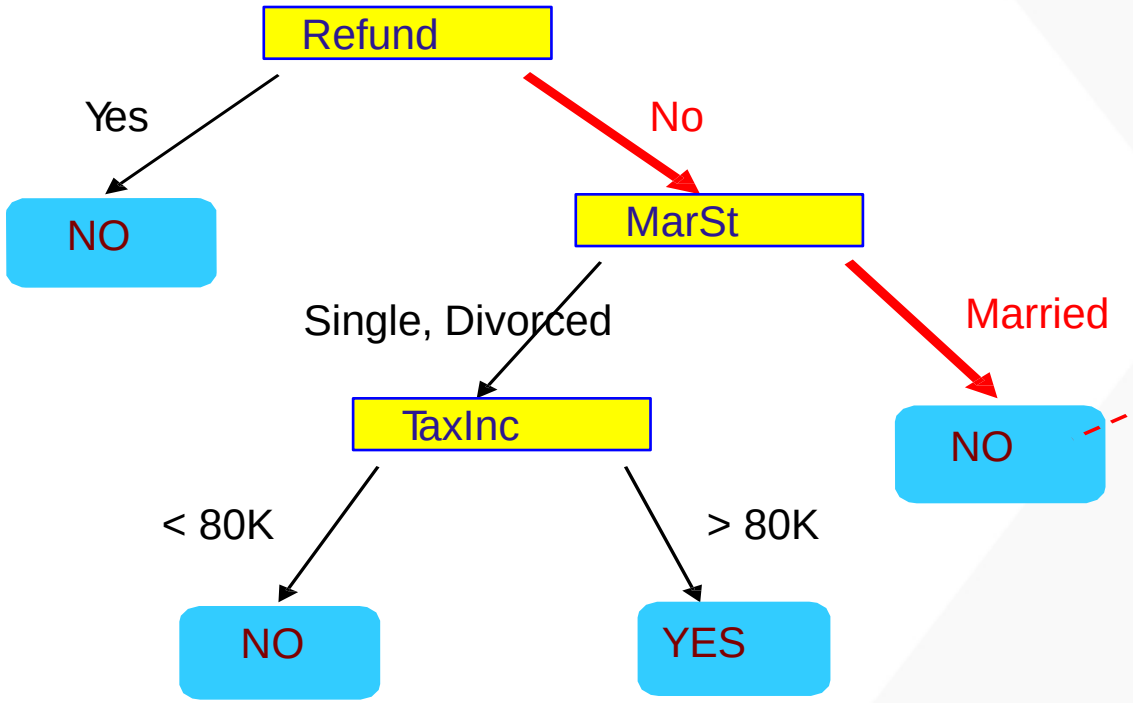
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assign Cheat to “No”

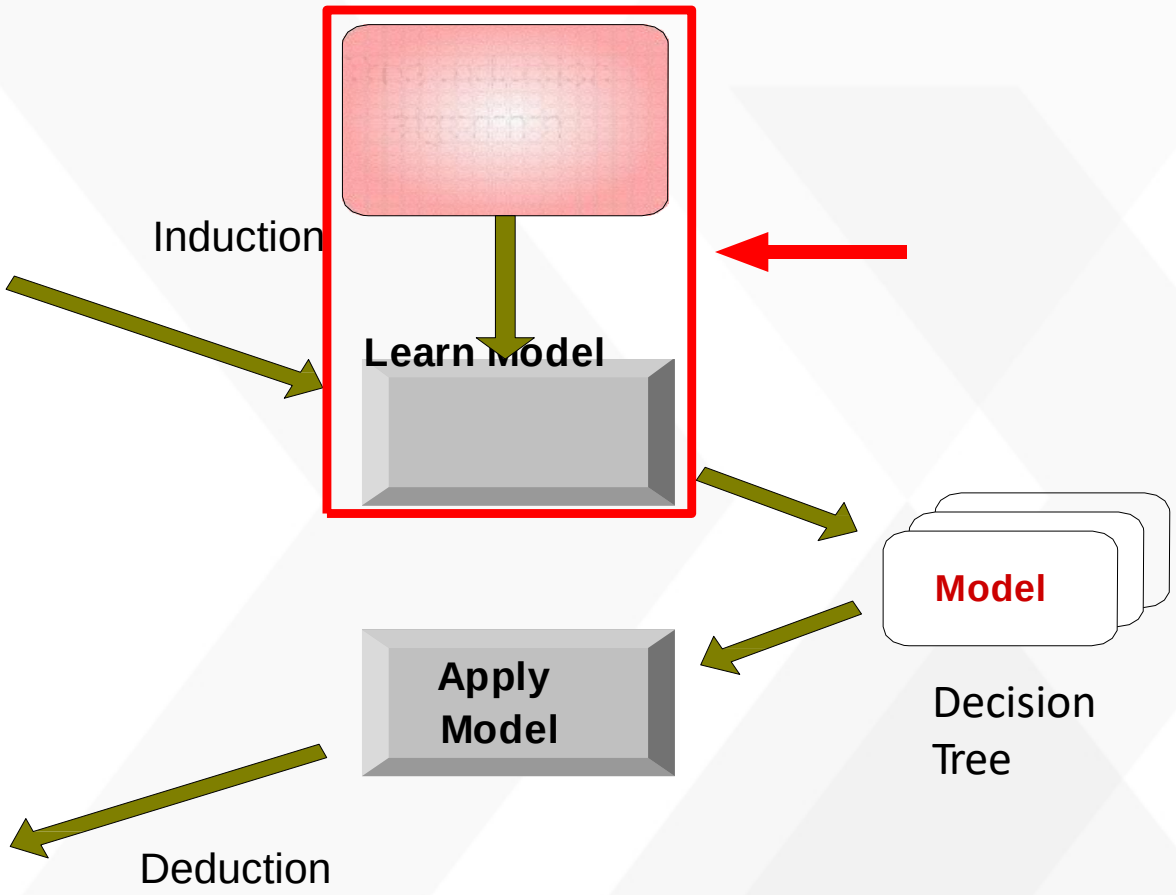
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



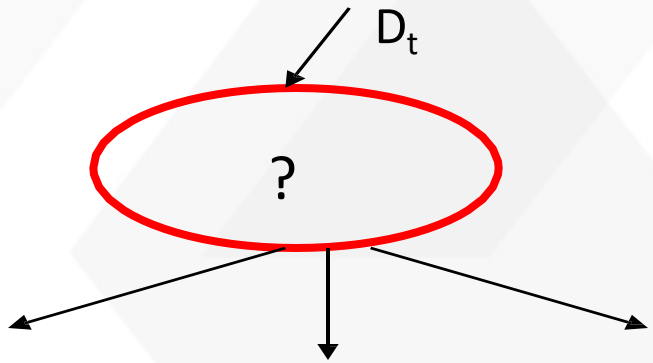
Decision Tree Induction

- **Many Algorithms:**
  - Hunt's Algorithm
  - CART
  - ID3, C4.5
  - SLIQ,SPRINT
  - ……

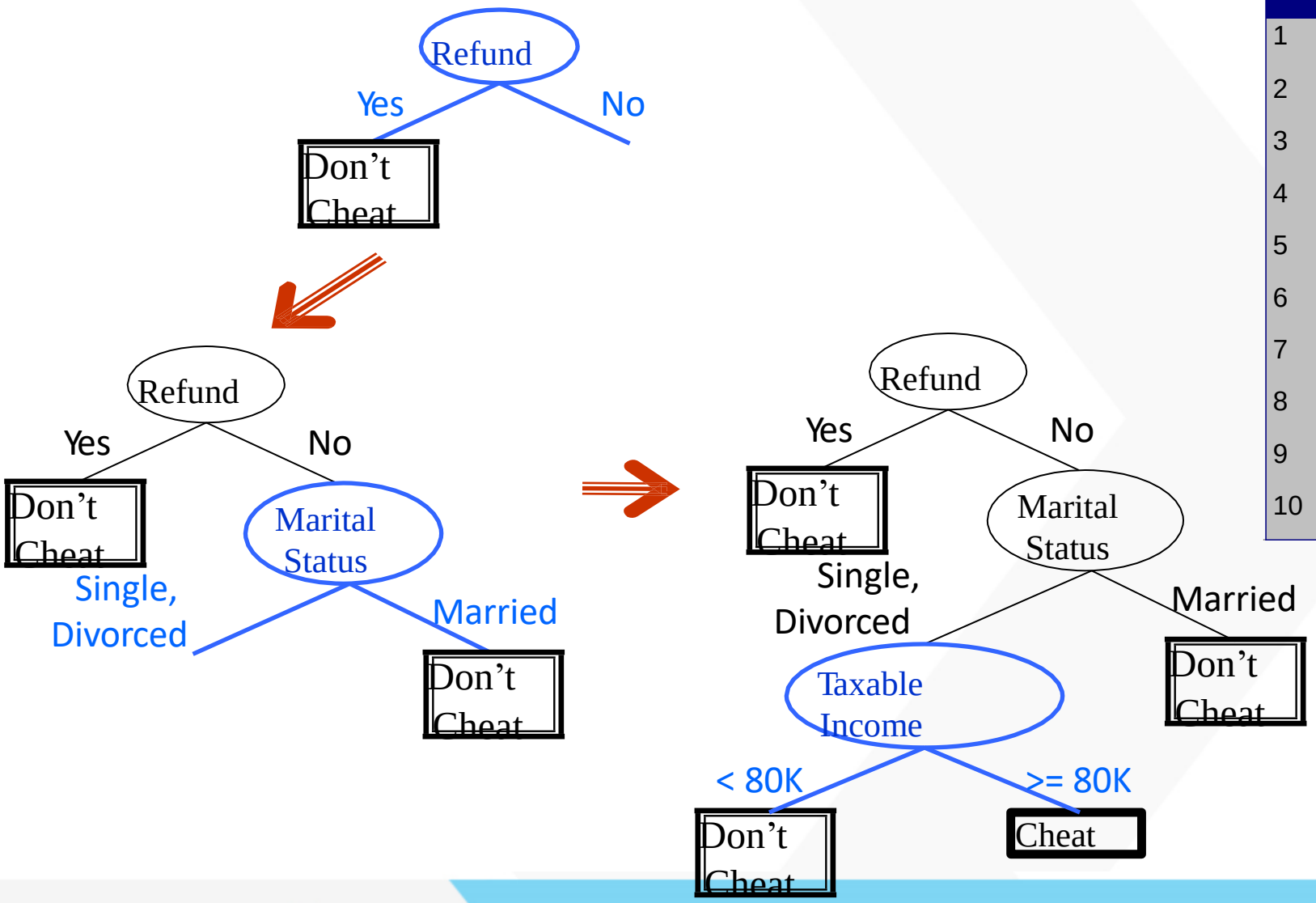
General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
- **General Procedure:**
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t contains records that belong to more than one class, use an attribute to split the data into smaller subsets. Recursively apply the procedure to each subset

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Hunt's Algorithm



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tree Induction

- **Greedy strategy**
  - Split the records based on an attribute test that optimizes certain criterion

- **Issues**
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

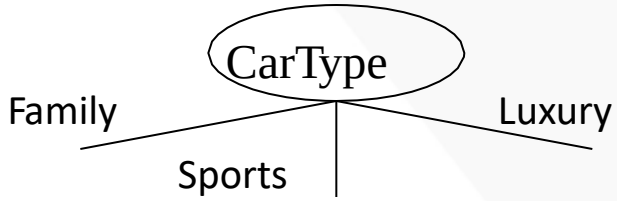
How to Specify Test Condition?

- **Depends on attribute types**
  - Nominal
  - Ordinal
  - Continuous

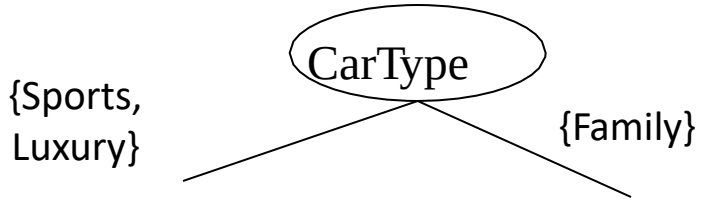
- **Depends on number of ways to split**
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

- **Multi-way split:** Use as many partitions as distinct values



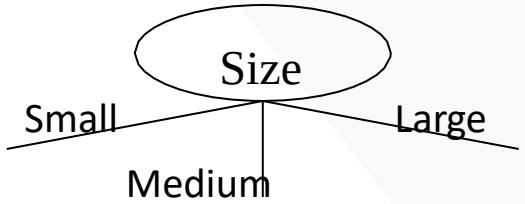
- **Binary split:** Divides values into two subsets
  Need to find optimal partitioning



OR

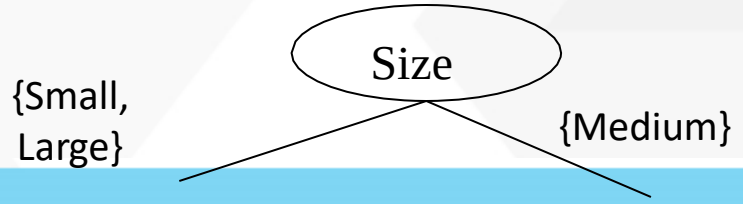
Splitting Based on Ordinal Attributes

- **Multi-way split:** Use as many partitions as distinct values.



- **Binary split:** Divides values into two subsets
  Need to find optimal partitioning

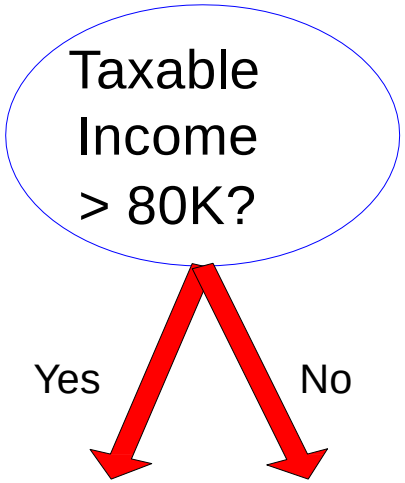
- What about this split?



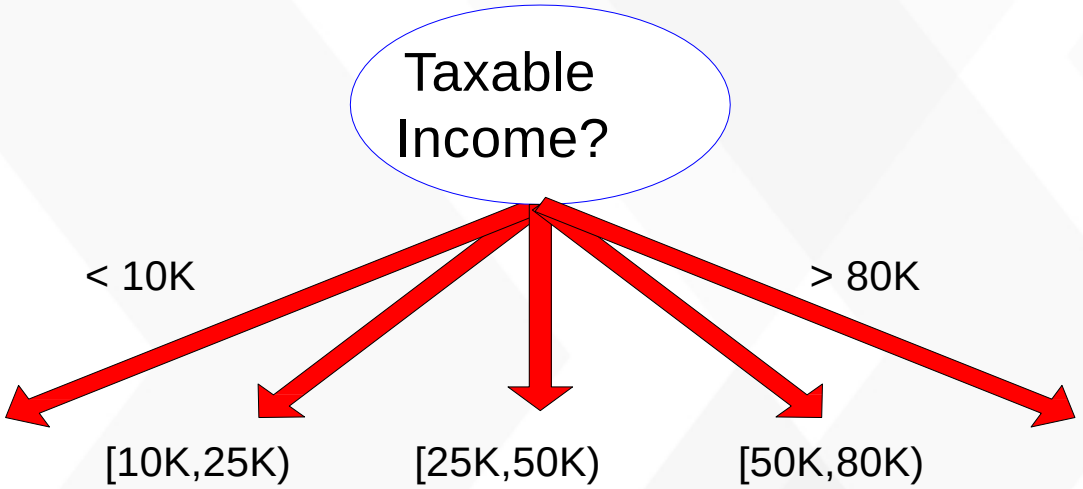
Splitting Based on Continuous Attributes

- Different ways of handling
  - **Discretization** to form an ordinal categorical attribute
  - **Binary Decision:** $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut
    - can be more computation intensive

Splitting Based on Continuous Attributes



(i) Binary split



(ii) Multi-way split

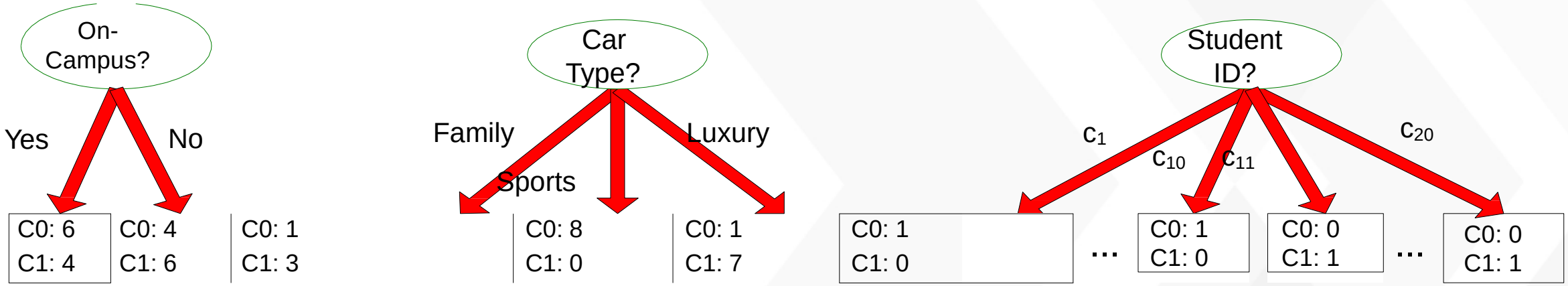
Tree Induction

- **Greedy strategy**
  - Split the records based on an attribute test that optimizes certain criterion.

- **Issues**
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1



Which test condition is the best?

How to determine the Best Split

- **Greedy approach:**
  - Nodes with homogeneous class distribution are preferred
- **Need a measure of node impurity:**

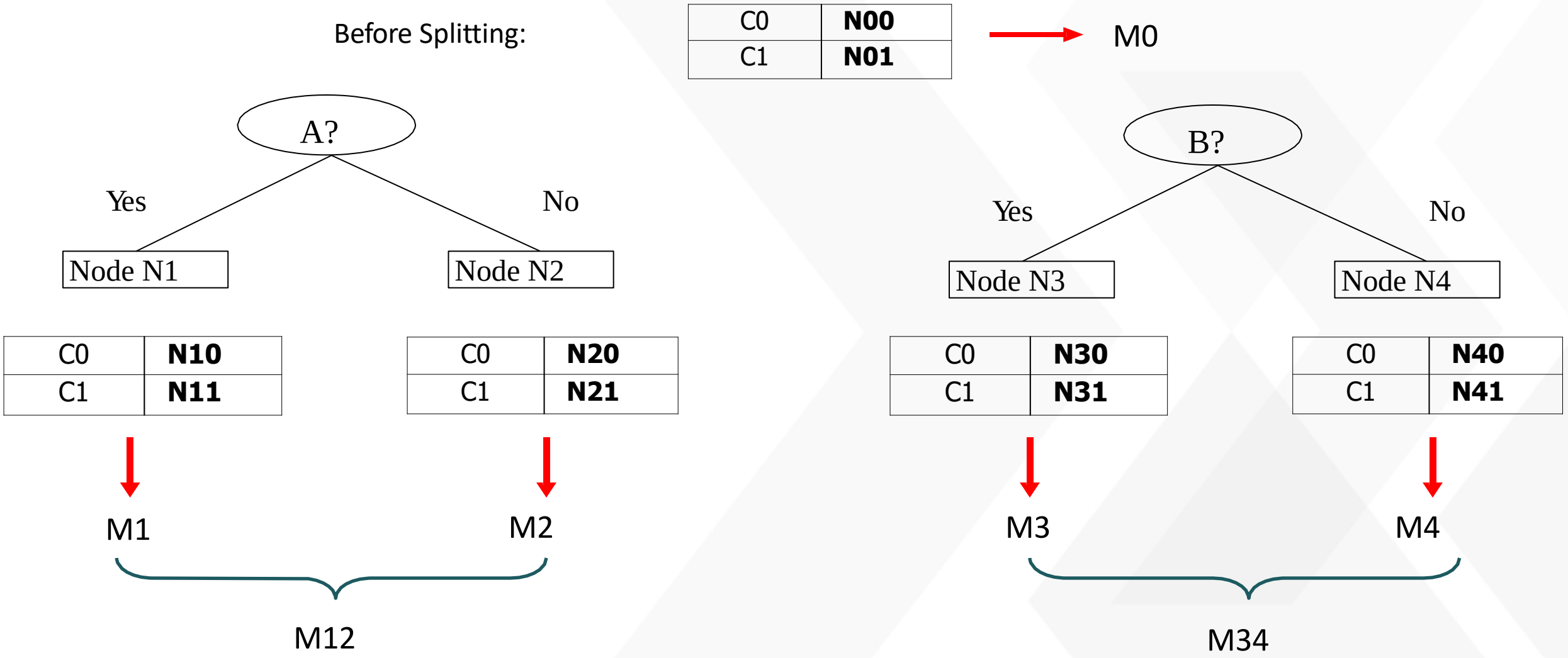
C0: 5 C1: 5

Non-homogeneous, High degree of impurity

C0: 9 C1: 1

Homogeneous, Low degree of impurity

How to Find the Best Split



Gain = M0 – M12 vs M0 – M34

Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error

Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

  - (NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

– Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information

– Minimum (0) when all records belong to one class, implying

- most useful information

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Examples for computing GINI

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

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

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

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

Gini = 1 – (1/6)² – (5/6)² = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)² – (4/6)² = 0.444

Splitting Based on GINI

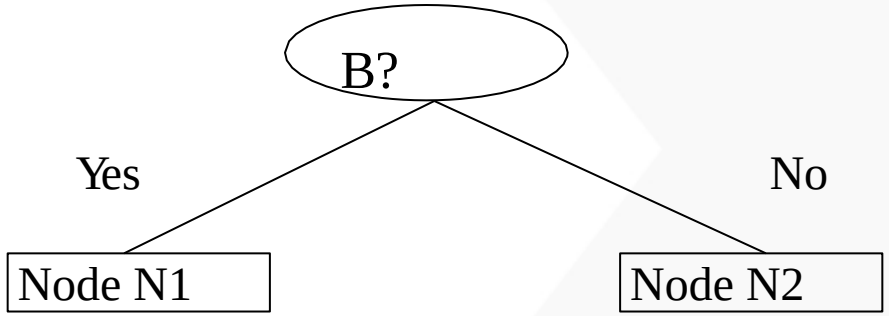
- Used in CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i, n = number of records at node p.

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for



	Parent
C1	**6**
C2	**6**
Gini = 0.500	

Gini(N1)
= 1 – $(5/7)^2$ – $(2/7)^2$
= 0.408

Gini(N2)
= 1 – $(1/5)^2$ – $(4/5)^2$
= 0.32

	N1	N2
C1	**5**	**1**
C2	**2**	**4**
Gini=0.333		

Gini(Children)
= 7/12 * 0.408 + 5/12 * 0.32
= 0.371

Entropy

- Entropy at a given node t:

$$Entropy(t) = -\sum p(j \mid t) \log p(j \mid t)$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  - Measures purity of a node
    - Maximum (log n_c) when records are equally distributed among all classes implying least information
    - Minimum (0.0) when all records belong to one class, implying most information

Examples for computing Entropy

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	**0**
C2	**6**

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

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

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

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Splitting Based on Information Gain

- **Information Gain:**

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions; n_i is number of records in partition i

  - Measures reduction in entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
  - Used in ID3 and C4.5

Splitting Criteria based on Classification Error

- Classification error at a node t :

$$Error(t) = 1 - \max P(i_i \mid t)$$

- Measures misclassification error made by a node.
  - Maximum (1 - $1/n_c$) when records are equally distributed among all classes, implying least interesting information
  - Minimum (0.0) when all records belong to one class, implying most interesting information

Examples for Computing Error

$$Error(t) = 1 - \max P(i_i \mid t)$$

C1	**0**
C2	**6**

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

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

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

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

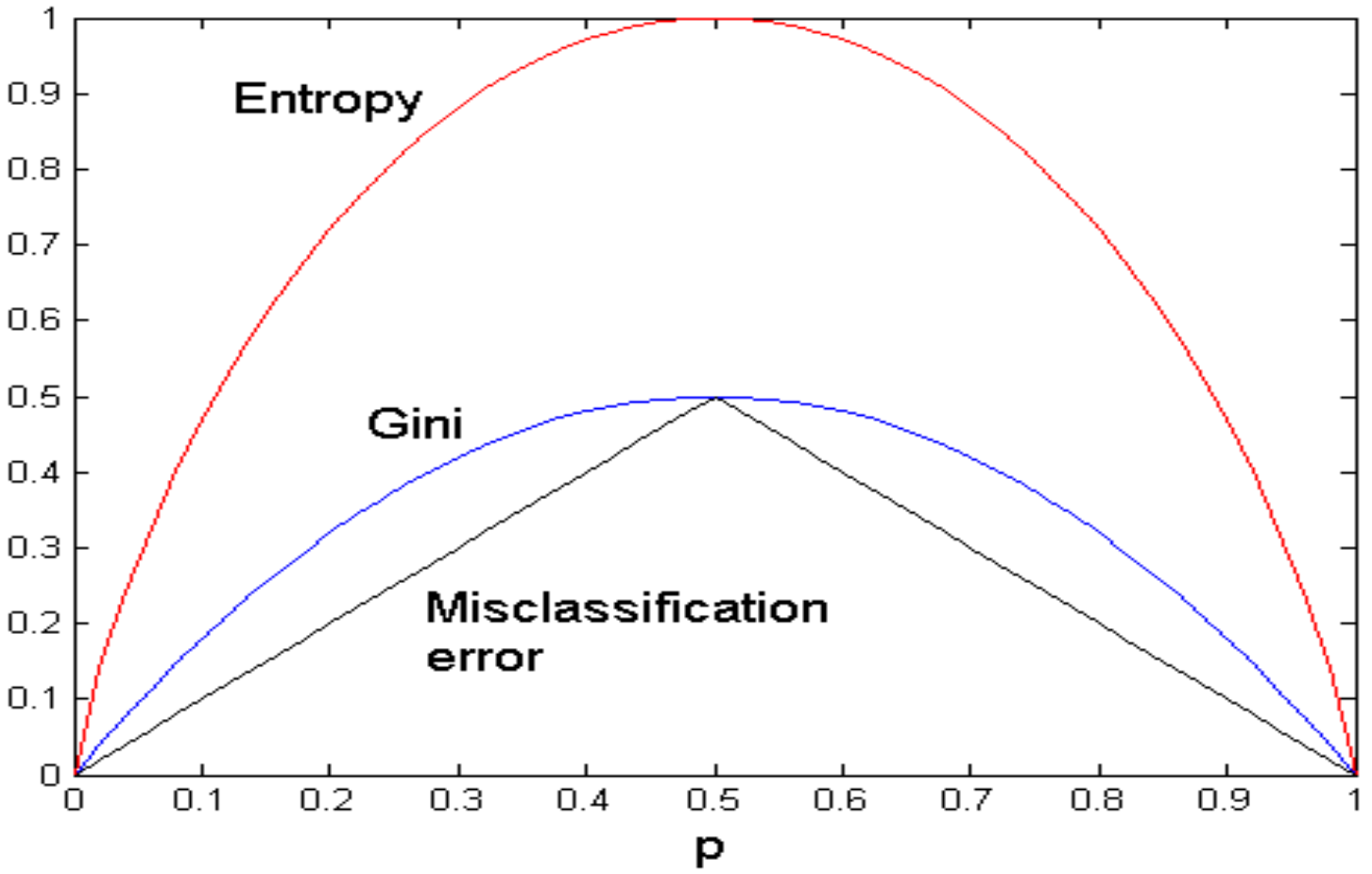
C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Comparison among Splitting Criteria

For a 2-class problem:



Tree Induction

- **Greedy strategy**
  - Split the records based on an attribute test that optimizes certain criterion.

- **Issues**
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

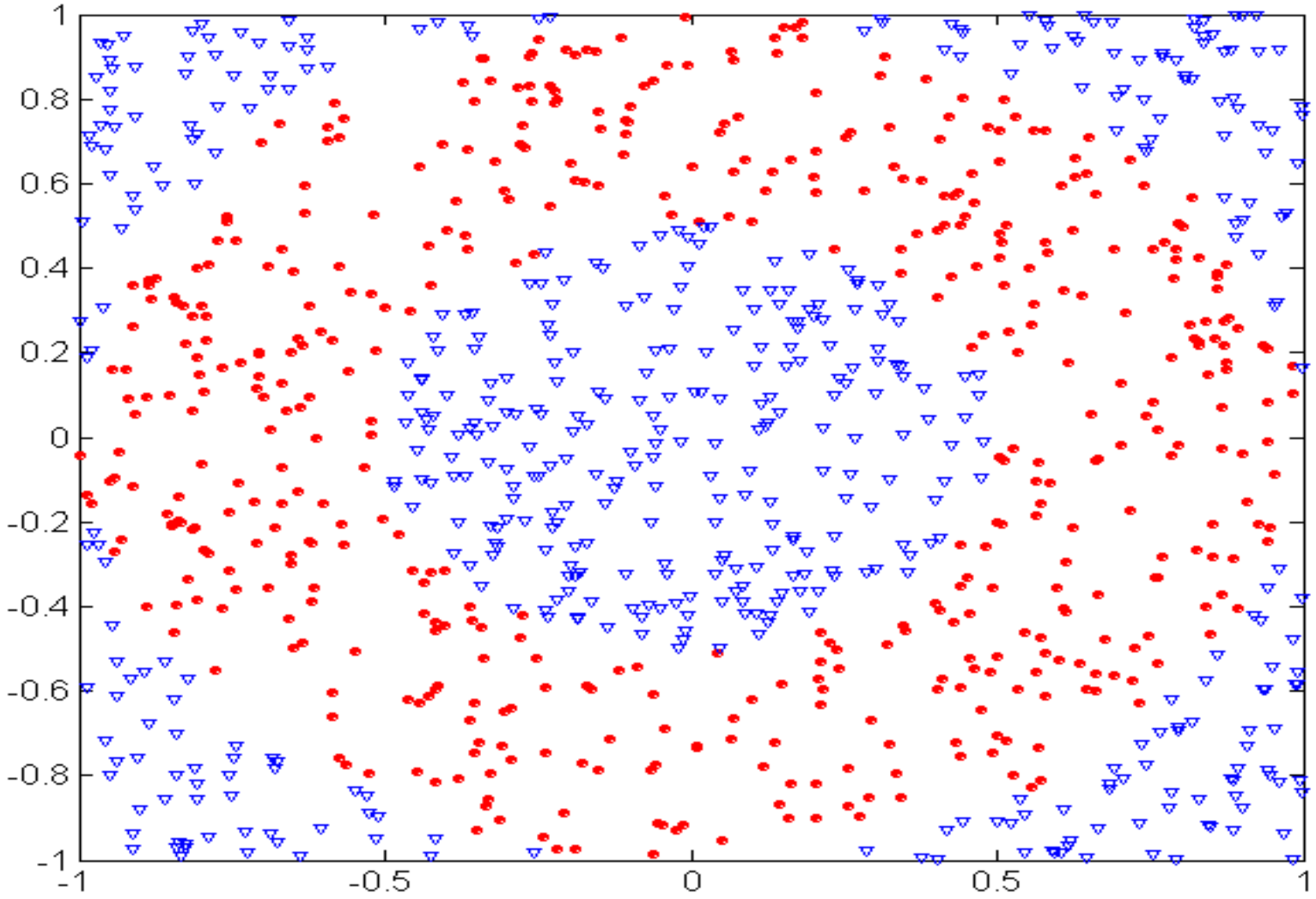
Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination (to be discussed later)

Decision Tree Based Classification

- **Advantages:**
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Accuracy is comparable to other classification techniques for many simple data sets

Underfitting and Overfitting (Example)



500 circular and 500 triangular data points.

Circular points:

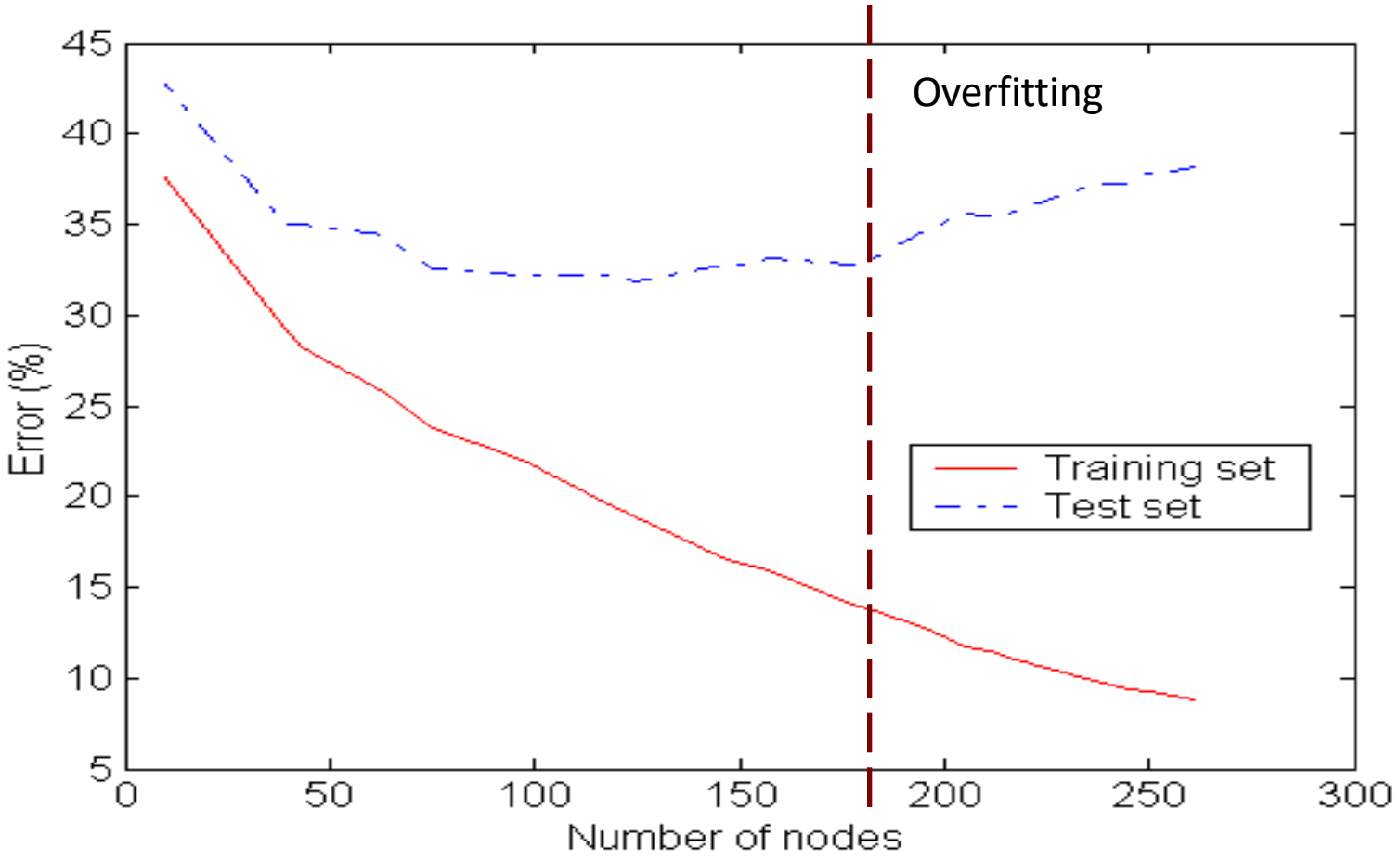
$0.5 \leq \text{sqrt}(x_1^2 + x_2^2) \leq 1$

Triangular points:

$\text{sqrt}(x_1^2 + x_2^2) > 0.5$ or

$\text{sqrt}(x_1^2 + x_2^2) < 1$

Underfitting and Overfitting



Occam's Razor

- Given two models of similar errors, one should prefer the simpler model over the more complex model
- For complex models, there is a greater chance that it was fitted accidentally by errors in data
- Therefore, one should include model complexity when evaluating a model

How to Address Overfitting

- **Pre-Pruning (Early Stopping Rule)**
  - Stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node:
    - Stop if all instances belong to the same class
    - Stop if all the attribute values are the same
  - More restrictive conditions:
    - Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).

How to Address Overfitting

- Post-pruning
  - Grow decision tree to its entirety
  - Trim the nodes of the decision tree in a bottom-up fashion
  - If generalization error improves after trimming,
    - replace sub-tree by a leaf node.
  - Class label of leaf node is determined from majority class of instances in the sub-tree

Handling Missing Attribute Values

- Missing values affect decision tree construction in three different ways:
  - Affects how impurity measures are computed
  - Affects how to distribute instance with missing value to child nodes
  - Affects how a test instance with missing value is classified

Computing Impurity Measure

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	?	Single	90K	Yes

Missing value

Before Splitting:

Entropy(Parent)

= -0.3 log(0.3)-(0.7)log(0.7) = 0.8813

	Class = Yes	Class = No
Refund=Yes	**0**	**3**
Refund=No	**2**	**4**
Refund=?	**1**	**0**

Split on Refund: Entropy(Refund=Yes) = 0

Entropy(Refund=No)

= -(2/6)log(2/6) – (4/6)log(4/6) = 0.9183

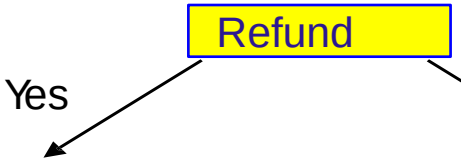
Entropy(Children)

= 0.3 (0) + 0.6 (0.9183) = 0.551

Gain = 0.9 × (0.8813 – 0.551) = 0.3303

Distribute Instances

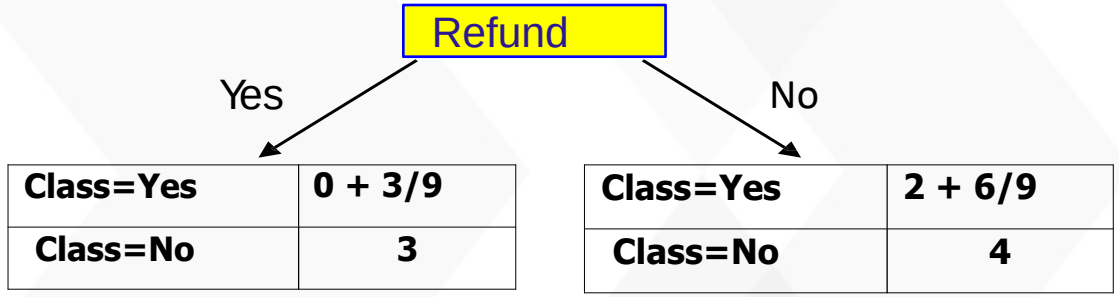
Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No



Class=Yes	0
Class=No	3

Cheat=Yes	2
Cheat=No	4

Tid	Refund	Marital Status	Taxable Income	Class
10	?	Single	90K	Yes



Class=Yes	0 + 3/9
Class=No	3

Class=Yes	2 + 6/9
Class=No	4

Probability that Refund=Yes is 3/9 Probability that

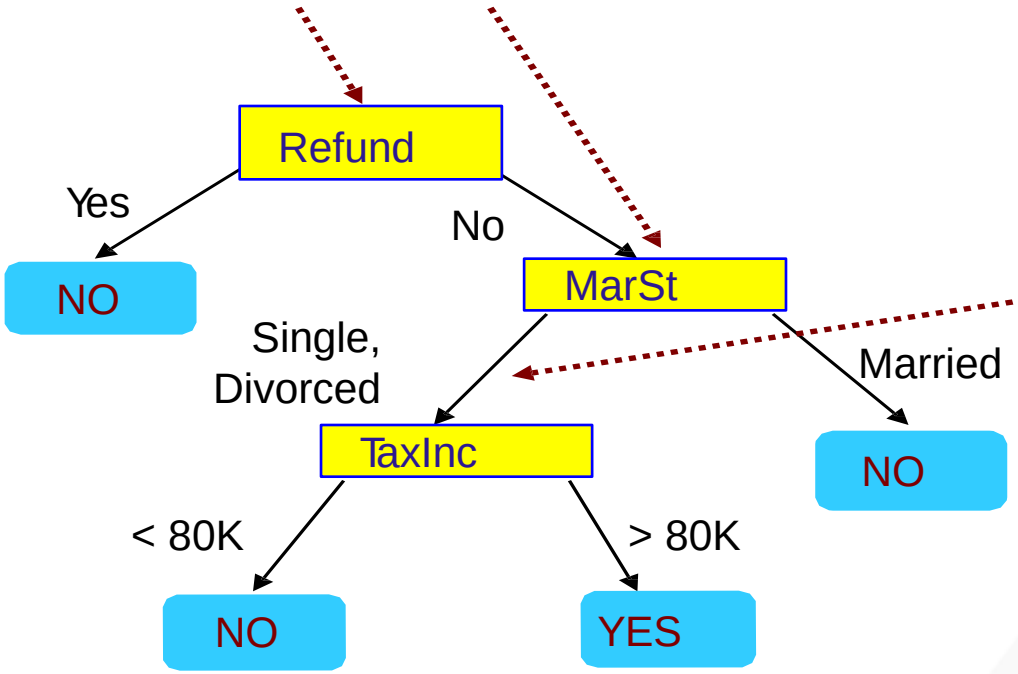
Refund=No is 6/9

Assign record to the left child with weight = 3/9 and to the right child with weight = 6/9

Classify Instances

New record:

Tid	Refund	Marital Status	Taxable Income	Class
11	No	?	85K	?



	Married	Single	Divorced	Total
Class=No	3	1	0	4
Class=Yes	6/9	1	1	2.67
Total	3.67	2	1	6.67

Probability that Marital Status = Married is 3.67/6.67

Probability that Marital Status ={Single,Divorced} is 3/6.67

Other Issues

- Data Fragmentation
- Search Strategy
- Expressiveness
- Tree Replication

Data Fragmentation

- Number of instances gets smaller as you traverse down the tree

- Number of instances at the leaf nodes could be too small to make any statistically significant decision

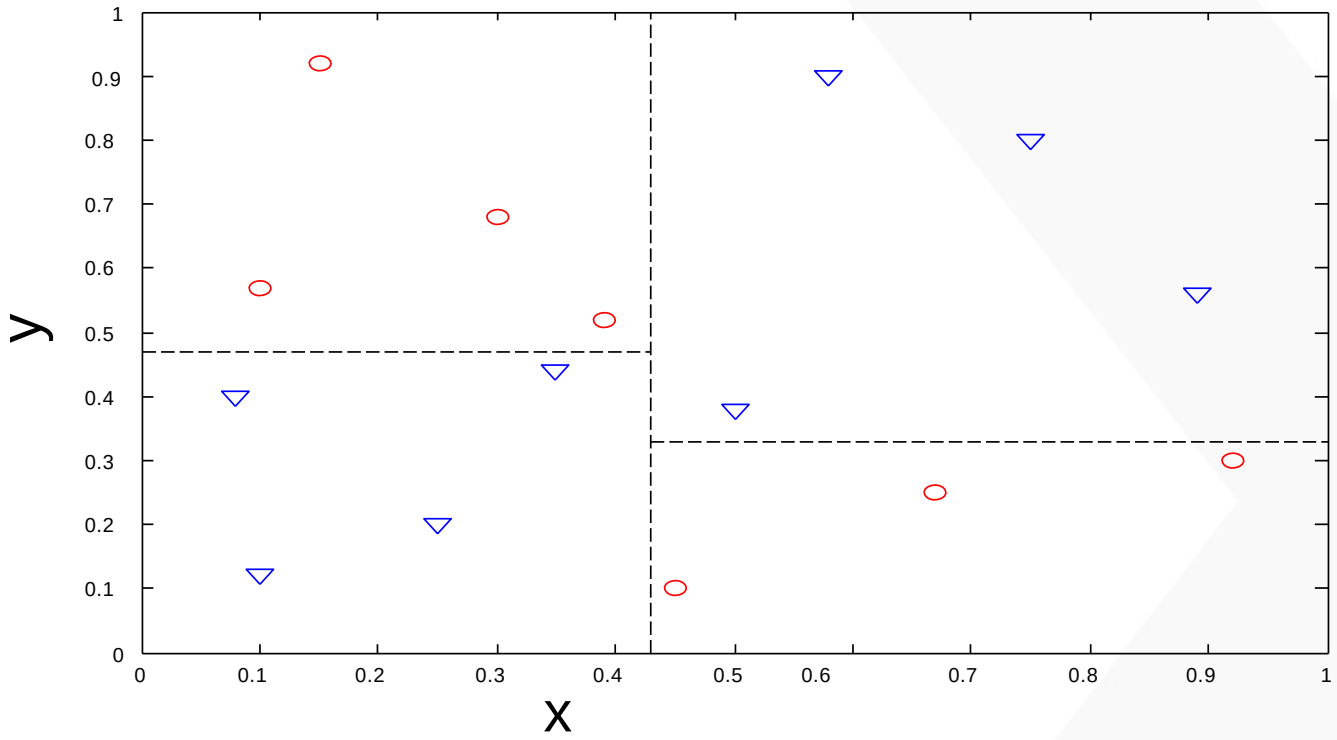
Search Strategy

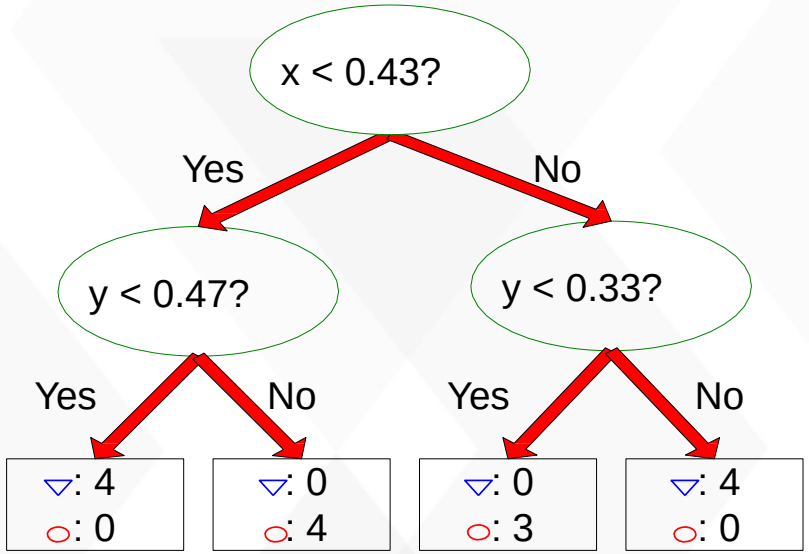
- Finding an optimal decision tree is NP-hard
- The algorithm presented so far uses a greedy, top-down, recursive partitioning strategy to induce a reasonable solution
- Other strategies?
  - Bottom-up
  - Bi-directional

Expressiveness

- Decision tree provides expressive representation for learning discrete-valued function
  - But they do not generalize well to certain types of Boolean functions
    - Example: parity function:
      - Class = 1 if there is an even number of Boolean attributes with truth value = True
      - Class = 0 if there is an odd number of Boolean attributes with truth value = True
    - For accurate modeling, must have a complete tree
- Not expressive enough for modeling continuous variables
  - Particularly when test condition involves only a single attribute at-a-time

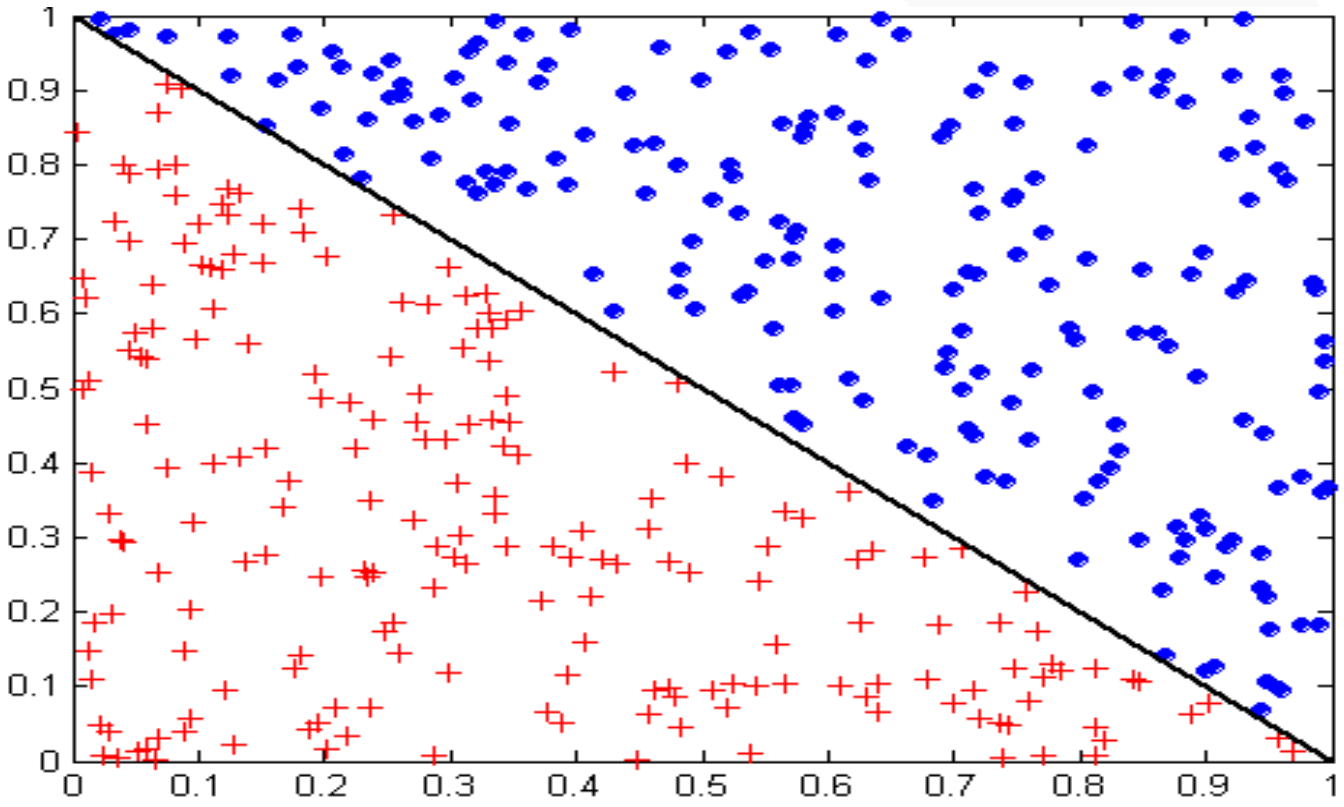
Decision Boundary

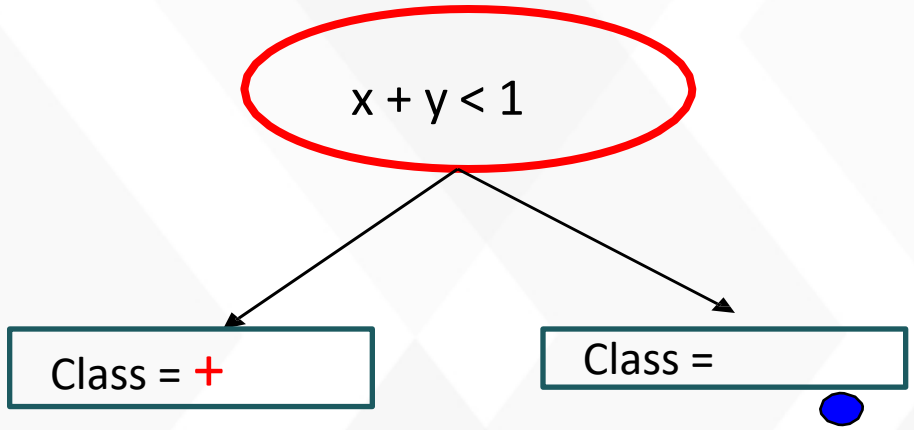




- Border line between two neighboring regions of different classes is known as decision boundary
- Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

Oblique Decision Trees





- Test condition may involve multiple attributes
- More expressive representation
- Finding optimal test condition is computationally expensive

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive) b: FN (false negative) c: FP (false positive) d: TN (true negative)

Metrics for Performance Evaluation

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

Methods of Estimation

- **Holdout**
  - Reserve 2/3 for training and 1/3 for testing
- **Random subsampling**
  - Repeated holdout
- **Cross validation**
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one
  - Leave-one-out: k=n
- **Stratified sampling**
  - oversampling vs undersampling
- **Bootstrap**
  - Sampling with replacement

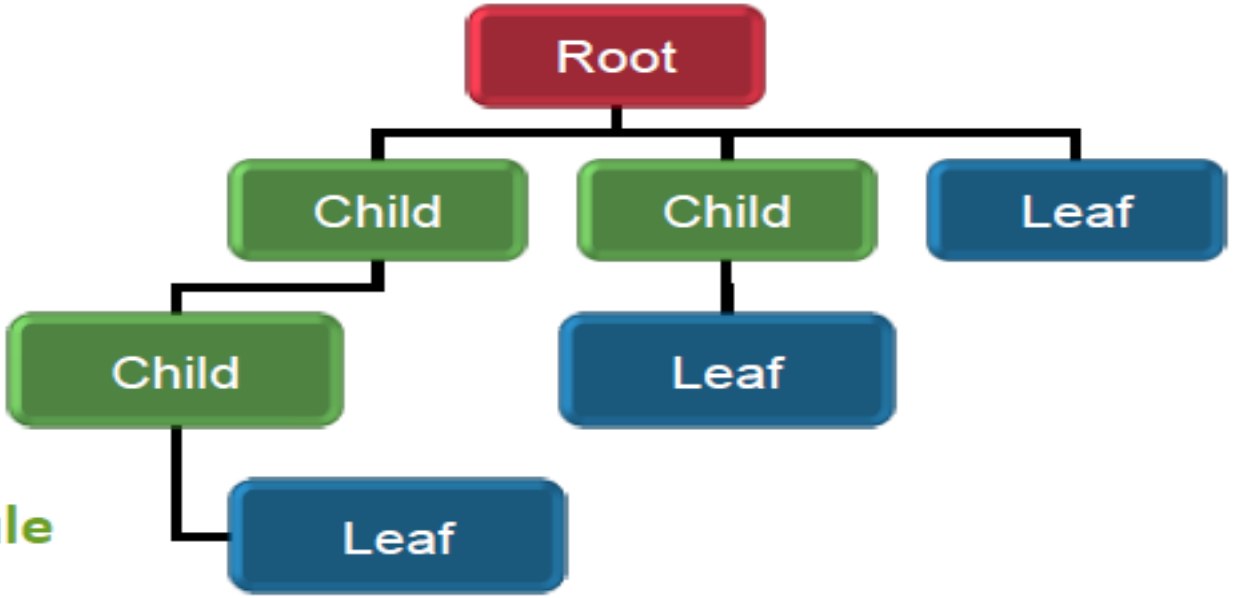
Take-away Message

- What's classificatioŶ?
- How to use decision tree to make predictions?
- How to construct a decision tree from training data?
- How to compute gini index, entropy, misclassification error?
- How to avoid overfitting by pre-pruning or post- pruning decision tree?
- How to evaluate classification model?

Objective Segmentation: Decision Trees

Decision Tree Vocabulary

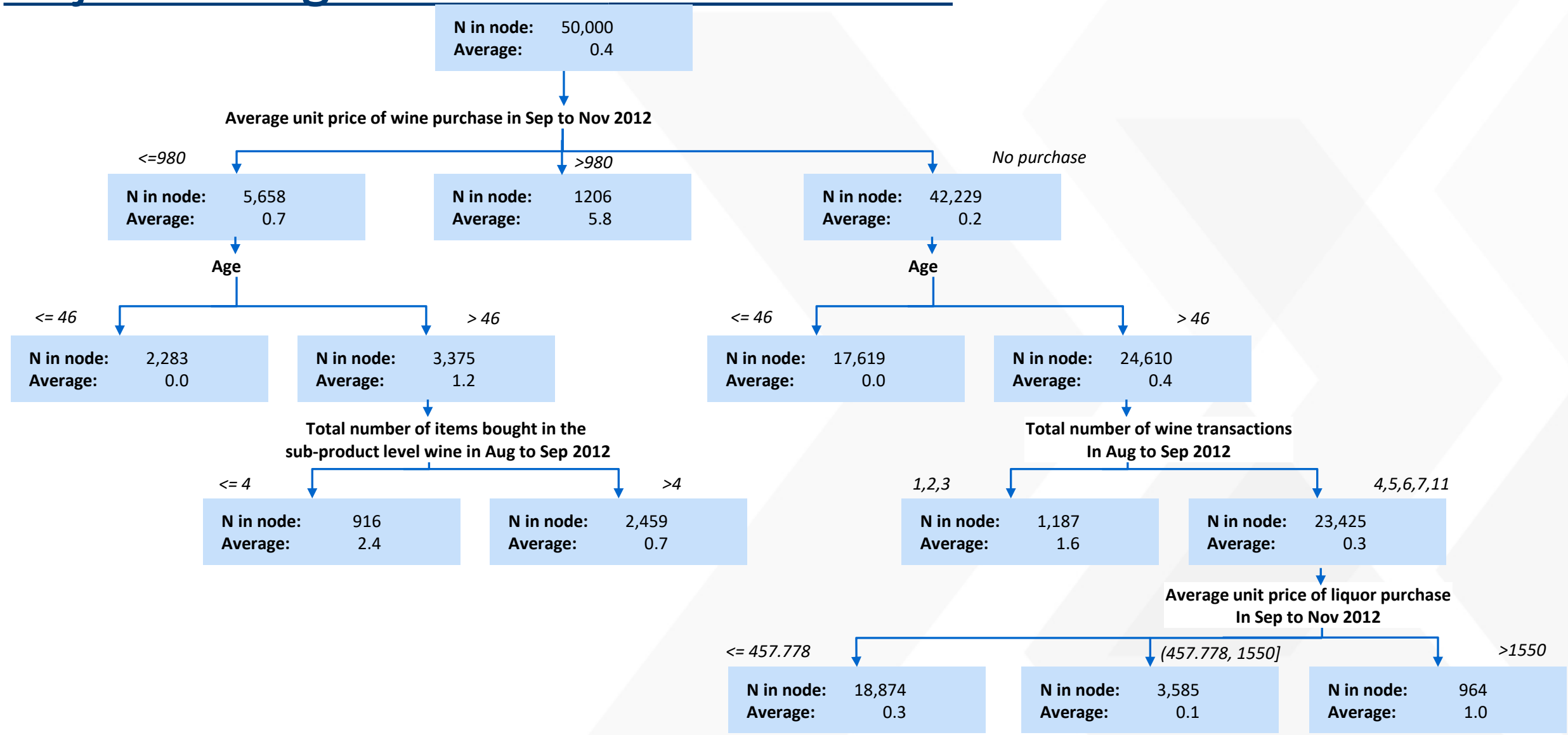
- Drawn top-to-bottom or left-to-right
- Top (or left-most) node = **Root Node**
- Descendent node(s) = **Child Node(s)**
- Bottom (or right-most) node(s) = **Leaf Node(s)**
- Unique path from root to each leaf = **Rule**



Decision Tree Types

- **Binary trees** – only two choices in each split. Can be non-uniform (uneven) in depth
- **N-way trees** or ternary trees – three or more choices in at least one of its splits (3-way, 4-way, etc.)

Objective Segmentation: Decision Trees



Decision Tree Example– business rules

Business rule statistics and description

Business Rules	Propensity to buy	% Customer	Description of the rule
Rule1	5.80	2. 5	Average unit price of wine purchase in Sep to Nov 2012 = >980
Rule2	2.40	1. 9	Average unit price of wine purchase in Sep to Nov 2012 = <=980; Age = >46; Total number of items bought in sub-product level wine in Aug to Sep 2012 = <=4
Rule3	1.60	2. 4	No wine purchase in Sep to Nov 2012; Age = >46; Total wine transactions in Sep to Nov 2012 = 1,2,3
Rule4	1.04	2. 0	No wine purchase in Sep to Nov 2012; Age = >46; Total wine transactions in Sep to Nov 2012 = 4,5,6,7,11; Average of unit price of liquor purchase in Sep to Nov 2012 = >1,550
Rule5	0.69	5. 0	Average unit price of wine purchase in Sep to Nov 2012 = <=980; Age = >46; Total number of items bought in sub-product level wine in Aug to Sep 2012 = >4
Rule6	0.31	38. 4	No wine purchase in Sep to Nov 2012; Age = >46; Total wine transactions in Sep to Nov 2012 = 4,5,6,7,11; Average of unit price of liquor purchase in Sep to Nov 2012 = <=457.778
Rule7	0.14	7. 3	No wine purchase in Sep to Nov 2012; Age = >46; Total wine transactions in Sep to Nov 2012 = 4,5,6,7,11; Average of unit price of liquor purchase in Sep to Nov 2012 = (457.778,1550]
Rule8		35. 9	No wine purchase in Sep to Nov 2012; Age = <=46
Rule9		4. 7	Average unit price of wine purchase in Sep to Nov 2012 = <=980; Age = <=46

Decision Trees: CHAID Segmentation

- CHAID- Chi-Squared Automatic Interaction Detector
- CHAID is a non-binary decision tree.
- The decision or split made at each node is still based on a single variable, but can result in multiple branches.
- The split search algorithm is designed for categorical variables.
- Continuous variables must be grouped into a finite number of bins to create categories.
  - A reasonable number of “equal population bins” can be created for use with CHAID.
  - ex. If there are 1000 samples, creating 10 equal population bins would result in 10 bins, each containing 100 samples.

- A Chi-square value is computed for each variable and used to determine the best variable to split on.

CHAID Algorithm

1. Select significant independent variable
2. Identify category groupings or interval breaks to create groups most different with respect to the dependent variable
3. Select as the primary independent variable the one identifying groups with the most different values of the dependent variable based on chi-square
4. Select additional variables to extend each branch if there are further significant differences

Introduction to Factor Analysis - PCA

Look at below Cricket Team Players Data

Player	Avg Runs	Total wickets	Height	Not outs	Highest Score	Best Bowling
1	45	3	5.5	15	120	1
2	50	34	5.2	34	209	2
3	38	0	6	36	183	0
4	46	9	6.1	78	160	3
5	37	45	5.8	56	98	1
6	32	0	5.10	89	183	0
7	18	123	6	2	35	4
8	19	239	6.1	3	56	5
9	18	96	6.6	5	87	7
10	16	83	5.9	7	32	7
11	17	138	5.10	9	12	6

Describe the players

- If we have to describe or segregate the players do we really need Avg Runs, Total wickets, Height, Not outs, Highest Score, Best bowling?
- Can we simply take
  - Avg Runs+ Not outs+ Highest Score as one factor?
  - Total wickets+ Height+ Best bowling as second factor?

Defining these imaginary variables or a linear combination of variables to reduce the dimensions is called PCA or FA

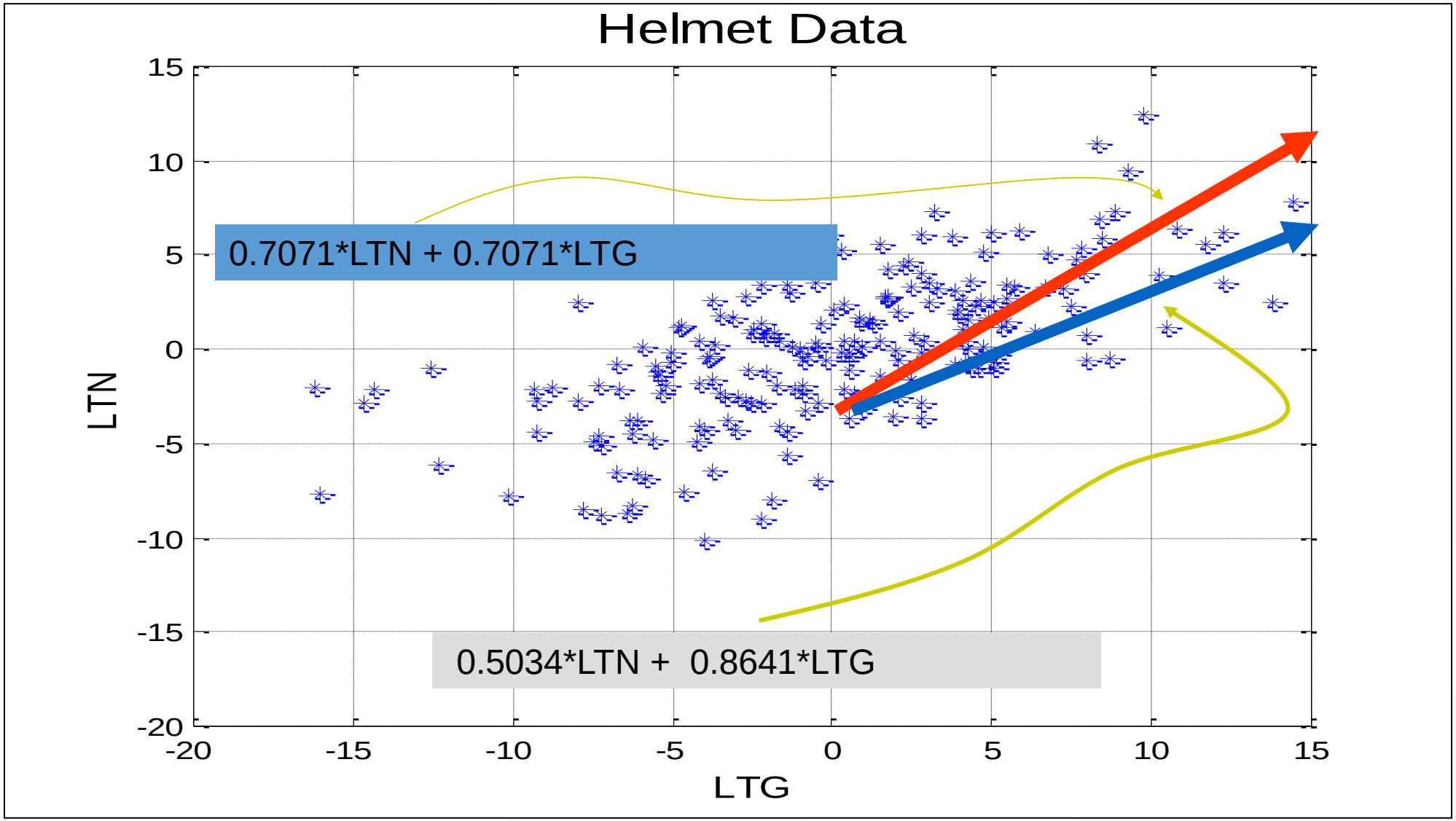
Look at below Cricket Team Players Data

Player	Avg Runs	Total wickets	Height	Not outs	Highest Score	Best Bowling
1	45	3	5.5	15	120	1
2	50	34	5.2	34	209	2
3	38	0	6	36	183	0
4	46	9	6.1	78	160	3
5	37	45	5.8	56	98	1
6	32	0	5.10	89	183	0
7	18	123	6	2	35	4
8	19	239	6.1	3	56	5
9	18	96	6.6	5	87	7
10	16	83	5.9	7	32	7
11	17	138	5.10	9	12	6

Purpose of PCA

- To find a linear combination of the original variables that has the largest variance possible.
- Need some restriction on the entries in the linear combination or problem is not well defined.
- Usually require sums of the squares of weights to be 1.

Example


Helmet Data
0.7071*LTN + 0.7071*LTG
0.5034*LTN + 0.8641*LTG
LTN
LTG
15
10
5
0
-5
-10
-15
-20
-20
-15
-10
-5
0
5
10
15

What is happening?

- Trying to find a direction where the physical scatter of points is most clearly “jutting out”
- This “diversity” may be just what you are looking for in your data
- Why would anyone want to find such directions?

Principal Components Regression

- Standard regression problem with response y and regressors X_1, X_2, …, X_p.
- X_1, X_2, …, X_p may be exactly collinear or nearly so.
- Least squares estimates of regression coefficients are not possible, or not reliable in that case.
- Can use Principal Components to address the problem.

Intelligent Index Formation

- May have answers to p questions, say X_1, X_2, …, X_p.
- And you may want to summarize these p responses with one number (“index”) that best captures the diversity in responses.
- E.g. is common to add the responses, or average them, perhaps being sensitive to questions that are reverse coded.
- Already should be clear to you that a simple averaging may not be the best way to summarize the original p questions.

Reduction of Dimension

- Often able to replace the original variables X_1, X_2, …, X_p with a few new variables, say, U_1, U_2, …, U_k where k is much smaller than p.

- By plotting the first two or three pairs of these new variables you can often see structure you wouldn't otherwise be able to see (e.g. clustering).

Interpretation

- In rarer cases the new variables, U_1, U_2, …, U_k, are interpretable and point to some new facet of the study.
- As you will see, however, one must be very careful with this use of Principal Components since it is a prime opportunity to go astray and over interpret.
- This is often where PCA is confused with Factor Analysis.

How Does PCA Work?

- Look for weights a_{11}, a_{12}, …, a_{1p} such that $U_1=a_{11}*X_1+a_{12}*X_2+\ldots+a_{1p}*X_p$ has the largest variance subject to the restriction that $(a_{11})^2+(a_{12})^2+\ldots+(a_{1p})^2=1$
- The numbers a_{11}, a_{12}, …, a_{1p} are called different things in different books. In SAS they are arrayed in a column and called the **first principal component "eigenvector".**
- If the X_i variables have had their individual means subtracted off, then the new variable $U_1$is called the **first principal component**, or in most texts, the **first principal component score.**

What's Next?

- Look for weights a_{21}, a_{22}, …, a_{2p} such that $U_2=a_{21}*X_1+a_{22}*X_2+\ldots+a_{2p}*X_p$ has the next largest variance subject to the restriction that $(a_{21})^2+(a_{22})^2+\ldots+(a_{2p})^2=1$

- The numbers a_{21}, a_{22}, …, a_{2p} are called different things in different books. In SAS they are arrayed in a column and called the **second principal component "eigenvector".**

- If the X_i variables have had their individual means subtracted off, then the new variable U_2 is called the **second principal component**, or in most texts, the **second principal component score.**

What's News?

- Any two arrays of weights will cross-multiply and sum to 0. Example: $(a_{11}\ a_{21})+(a_{12}a_{22})+\ldots+(a_{1p}\ a_{2p})=0$
- Same as saying: any two of the new variables will be uncorrelated. Example: $corr(U_1,U_2)=0$.

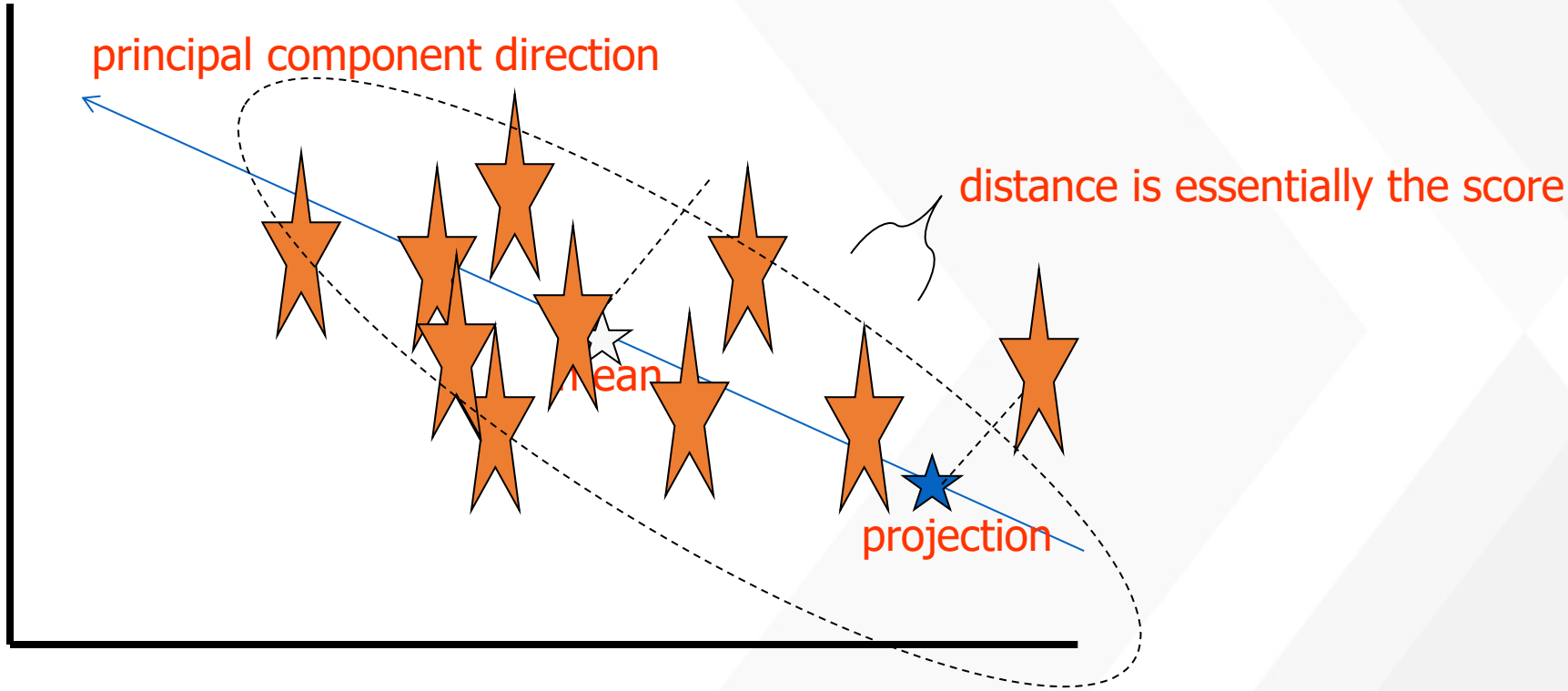
How Far Does This Go?

- Until the original data are described adequately.
- We will look at two or three criteria for how many of these scores to construct. We'll start with our common sense.
- Most of the time it is not as hard as it might sound. Basically, we will look at "how much variance" in the original data is summarized by each new component variable.

Two Basic Constructs

- Weights (used “a” to denote).
- Weights arrayed in columns and called “eigenvectors” on SAS output.
- Weights come from looking at all pairwise covariances associated with the original p variables.
- Scores (used “u” to denote).
- Scores called “principal components” and are the new variables.
- Typically use Weights for interpretation and development of subscales.
- Typically use Scores for clustering and as a substitution for the original data.

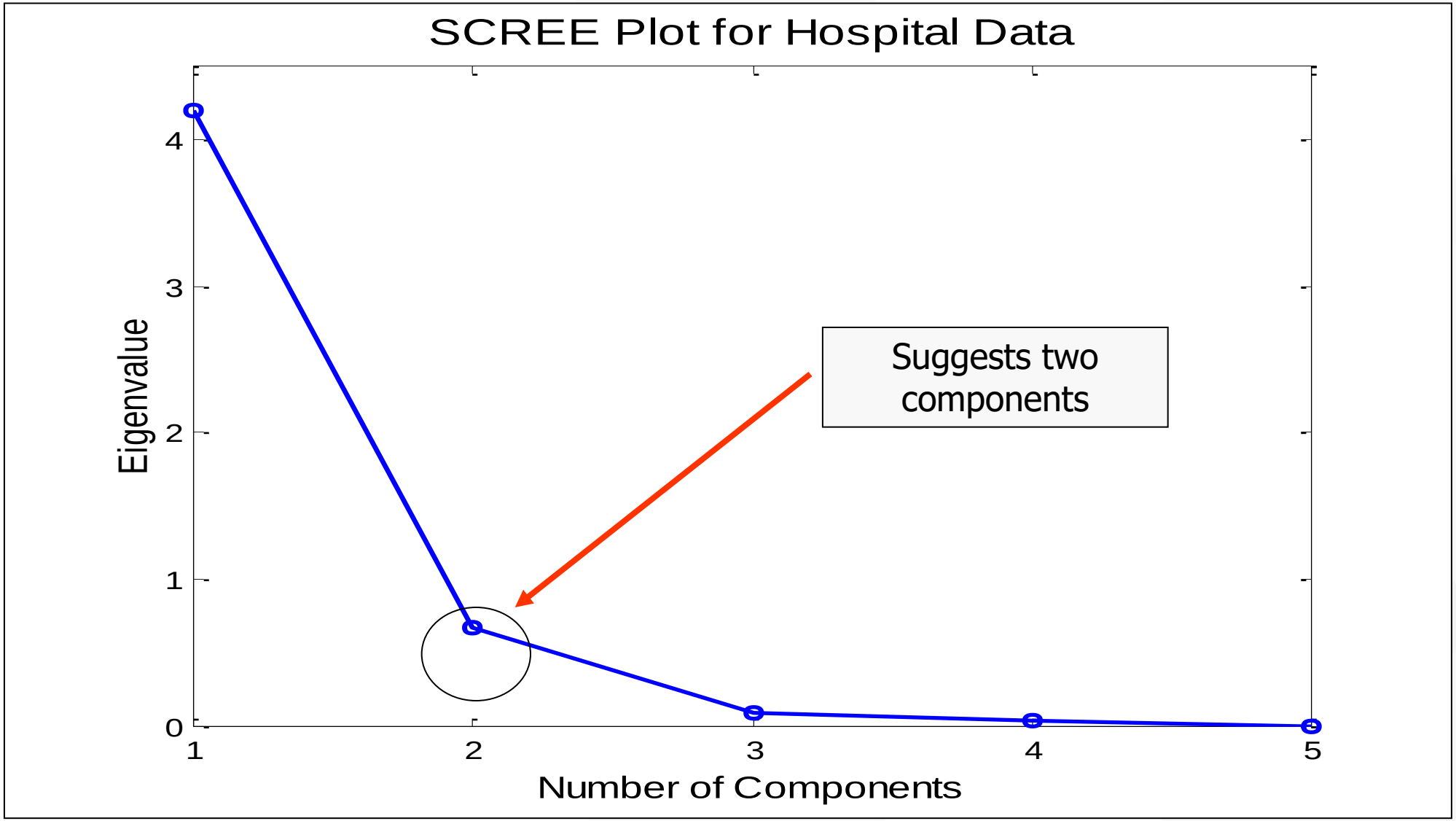
Geometry


principal component direction
distance is essentially the score
Mean
projection

Recall

	Eigenvalue	Difference	Proportion	Cumulative
1	4.19711750	3.52963341	0.8394	0.8394
2	0.66748410	0.57285125	0.1335	0.9729
3	0.09463284	0.05392125	0.0189	0.9918
4	0.04071159	0.04065762	0.0081	1.0000
5	0.00005397		0.0000	1.0000

Scree Plots


SCREE Plot for Hospital Data
Eigenvalue
0
1
2
3
4
1
2
3
4
5
Number of Components
Suggests two components

Loadings

Definition

Component loadings are the ordinary product-moment correlation between each original variable and each component score.

Interpretation

By looking at of component loadings one can ascertain which of the original variables tend to "load" on a given new variable. This may facilitate interpretations, creation of subscales, etc.

Q&A

Contact us

Visit us on: http://www.analytixlabs.in/

For course registration, please visit: http://www.analytixlabs.co.in/course-registration/

For more information, please contact us: http://www.analytixlabs.co.in/contact-us/

Or email: info@analytixlabs.co.in

Call us we would love to speak with you: (+91) 88021-73069

Join us on:

Twitter - http://twitter.com/#!/AnalytixLabs

Facebook - http://www.facebook.com/analytixlabs

LinkedIn - http://www.linkedin.com/in/analytixlabs

Blog - http://www.analytixlabs.co.in/category/blog/