<table>
<tr><th colspan="2">Classification</th></tr>
<tr><td colspan="2">- Split data accord. to some criterion, classify new (o) accord. to same criterion
- P(new (o) belong to each group) calc.
- New (o) assigned to group with highest prob.</td></tr>
<tr><th>Discriminant Analysis (Supervised learning)</th><th>Cluster Analysis (Unsupervised learning)</th></tr>
<tr><td>Groups specified (can identify from available data)</td><td>Groups not specified, need to find from data itself
Find hidden assoc. structures within unlabeled data</td></tr>
<tr><th>Logistic regression</th><th>Clustering</th></tr>
<tr><td>Training data: data used to build model
Misclassification: predict wrongly (Goal: min. this error)</td><td>Survival analysis: Failures of Samsung phones</td></tr>
<tr><td>Quote prob. rather than binary 0-1 decision
Labels of response variable known</td><td>Group similar objects but finds degree and nature of similarity by itself</td></tr>
<tr><td></td><td>- Preliminary classifier: rough labelling of (o) before thorough model-based analysis attempted
- Determine groups for discriminant analysis</td></tr>
<tr><td></td><td>K-means clustering
1. Choose value of k, guess for k cluster centroids
2. Compute distance of each pt from k centroids (Assign each pt. to cluster with closest centroid)
3. Recompute centroids (cluster mean) for each updated cluster
4. Con’t. step 2 & 3 until composition of clusters don’t change (i.e. convergence)
5. Final k clusters</td></tr>
<tr><td></td><td>Choice of k
- Min. WSS (Within sum of squares) of clusters
- Measures sum of ttl. dist. of (o) in each cluster from their respective centroids</td></tr>
<tr><th colspan="2">Decision Tree</th></tr>
<tr><td colspan="2">- Uses a tree structure to create a rule for classifying data
- Sequence of decisions specifying consequences
- Used for prediction</td></tr>
<tr><td>Build:
1. Full data with all its var. and (o)
<u>Sequence of if-then statements:</u>
2. Split data into groups
3. Each group split into subgroups
4. Splitting con’ts till certain stopping rule is reached or each (o) becomes a group of its own</td><td></td></tr>
<tr><td>Classification Tree</td><td>Regression Tree</td></tr>
<tr><td>- o/p var.s usually applied to be categorical in nature
- binary decisions like ‘yes’ or ‘no’</td><td>- o/p var.s can be numeric or cts</td></tr>
</table>

Association rules (Unsupervised learning)

- Uncover r/s of form X → Y: when X (o), item Y (o)
- Descriptive, not a predictive method
- Itemsets: items bought in the same transaction, webpages visited in the same sitting etc.
- k-itemset: { item 1, item 2, … , item k }

Apriori algorithm

- Given collection of items, explores all subsets of items
- Provides subsets which appear more than some pre-defined frequency

- Need transaction database D, min. length support threshold , max. length an itemset could reach N (optional)

1. Support	Given itemset L, ~ of L: proportion of transactions that contain L (0 < support < 1)
2. Frequent itemset	~ has items that appear tgt. often enough (satisfy a min. support criterion, usually set at 0.5)
3. Downward closure property (Apriori property)	If itemset considered frequent, any subset of freq. itemset also frequent.

Steps:

A) Bottom-up approach: Starting from 1-itemset, merges pairs of current itemsets to create new itemsets
B) Each stage: Algorithm checks if all itemsets satisfy min. support criterion
(If not, dropped and not considered in rest of algorithm)
C) Process continues till it runs out support or itemsets reach a predefined max. length

SQL (Structured Query Language)

In-database analytics: describes the processing of data within its repository
😊: database can update regularly → produce most up-to-date results
: Eliminate need to move data from place to place
: Well-protected/ Security → need to extract to another portal to make changes/ perform analysis

Relational database: part of RDBMS (Relational Database Management system)
😊: organizes data in tables with established r/s between tables
: splitting data into tables → no need to store entire table of info
: can change a specific part → no need to change everything and store in same place

1. Smaller memory to handle them
2. Various tables stored on diff. machines
3. Changes & corrections easily made
4. Relational database: number of duplicates reduced

Parallel Computation

Distributed: ~ of computation happens at processor level
: compilers optimised to distribute computation whenever

Parallel: run a function repeatedly in diff. processors
: snowFT → run full function in parallel

Residual bootstrap

Given dataset (x1, x2, … xn)

- Calculate mean, var.
- Resample the data of size n with replacement (means pts can be repeated)
- Recalc. Data
- Do for a lot of times (thousands?)

Can show that sample distribution will be close to the true distribution of the r.v. looking at