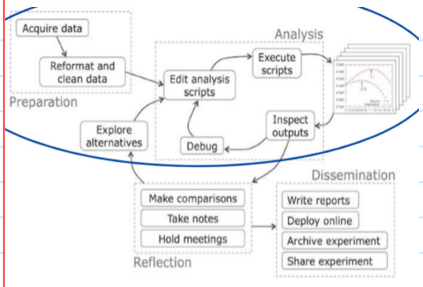


Nominal Data: unordered category data. coded in numeric form but have no mathematical interpretation , just labelling to denote categories. Central Tendency -> Mode
Dichotomous Data: type of nominal data, only has two possible values. (binary or Boolean variables)
Ordinal Data: Ordered categorical data in which there is strict monotonic order. Central tendency -> mode or median. Dispersion estimated by IQR. Mean not definable from ordinal set.
Interval Data: Interval scales give info on order, have equal intervals, No true zero , Addition is defined. Eg. Celsius temperature. Central tendency -> mode,median,mean (also for ratio)
Ratio Data: True value of zero and represents the total absence of the variable being measured. Eg. Kelvin (100K is twice of 50K). Values encode diff, Zero defined, Multiplication defined.

$\frac{\sum X_i}{N}$ $\frac{\sum (X_i - mean)^2}{N-1}$ $\sqrt{variance}$

Histogram (Illustrates tendency, categories differentiated)For Nominal data, can read mode. Dispersion: Counts/Distribution % For ordinal data, Dispersion: + min/max/range and percentiles.
Scatterplot: For ratio and interval data. Dispersion: + stdev/variance
BoxPlot - used when data is skewed. Values outside fence are outliers.
Correlation statistics measuring dependence - Pearson's r for two normally distributed variables. Spearman's rho for ratio and ordinal data. (rank-order correlation). Kendall's tau for ordinal variables.

```
CREATE TABLE Student
(
    sid         INTEGER         PRIMARY KEY,
    name        VARCHAR(20)     NOT NULL,
    gender      CHAR            CHECK (gender IN ('M','F','T')),
    birthday    DATE            NULL,
    country     VARCHAR(20),
    level       INTEGER         DEFAULT 1 CHECK (level BETWEEN 1 and 5)
);
```

Data Cleaning: Extract->Data Processing(Transform/clean)-> load to Data/metadata storage-> Feed to end user. 1.Type and name conversion, 2.filtering of missing or inconsistent data, 3.unifying semantic data representations, 4.matching of entries from different sources. Also, rescaling and optional dimensionality reduction. For 1 and 2 : clean() function like int() creates integer, float() creates floating point object, date/time.strptime() creates datetime objects from strings, Filter missing / wrongly formatted data and replace with default value

- Insertion of new data into a table / relation
  - Syntax:
    INSERT INTO *table* ["("*list-of-columns*")"] VALUES "("* list-of-expression *")"
  - Example:
    INSERT INTO Students (sid, name) VALUES (53688, 'Smith')
- Updating of tuples in a table / relation
  - Syntax:
    UPDATE *table* SET *column*"="*expression* {","*column*"="*expression*}
    [WHERE *search_condition*]
  - Example:
    UPDATE students
    SET gpa = gpa - 0.1
    WHERE gpa >= 3.3
- Deleting of tuples from a table / relation
  - Syntax:
    DELETE FROM *table* [WHERE *search_condition*]
  - Example:
    DELETE FROM Students WHERE name = 'Smith'

- List all details of the first three measurements including sensor data.

```
SELECT *
  FROM Measurement JOIN Sensor USING (sensor)
 LIMIT 3;
```

- Find the site name, commence date and organisation name of all stations :

```
SELECT sitename, commence, organisation
  FROM Station JOIN Organisation
  ON orgcode = code;
```

- GROUP BY Example:
  - What was the average mark of each course?

```
SELECT uos_code as unit_of_study, AVG(mark)
  FROM Assessment
GROUP BY uos_code
```

- HAVING clause: can further filter groups to fulfil a predicate
  - Example:

```
SELECT uos_code as unit_of_study, AVG(mark)
  FROM Assessment
GROUP BY uos_code
  HAVING AVG(mark) > 10
```

- Produces the cross-product ***Station x Sensor***

```
SELECT *
  FROM Station, Organisation;
```

- Find the site name, commence date and organisation name of all stations :

```
SELECT sitename, commence, organisation
  FROM Station S, Organisation O
 WHERE S.orgcode = O.code;
```

- Example 1: Which station commence after 1900-1-1?

```
SELECT siteName, commence, orgcode
  FROM station
 WHERE commence > '1900-1-1';
```

- Example 2: How many measurements we have done?

```
SELECT COUNT(*) FROM Measurement;
```

- Example 3: List top five measurements ordered by date in descending order.

```
SELECT * FROM Measurement ORDER BY date DESC limit 5;
```

SQL Aggregate Function	Meaning
COUNT(*attr*) ; COUNT(*)	Number of *Not-null-attr* ; or of all values
MIN(*attr*)	Minimum value of *attr*
MAX(*attr*)	Maximum value of *attr*
AVG(*attr*)	Average value of *attr* (arithmetic mean)
MODE() WITHIN GROUP (ORDER BY *attr*)	mode function over *attr*
PERCENTILE_DISC(0.5) WITHIN GROUP (ORDER BY *attr*)	median of the *attr* values
…	…

SQL Statement	Meaning
SELECT COUNT(*) FROM *T*	count how many tuples are stored in table *T*
SELECT * FROM *T*	list the content of table *T*
SELECT * FROM *T* LIMIT *n*	only list *n* tuples from a table
SELECT * FROM *T* ORDER BY *a*	order the result by attribute *a* (in ascending order; add DESC for descending order)

Types of statistical studies - Observational Studies (follow sample for future outcomes), Retrospective studies (Collect info about sample on specific outcomes that have happened)

P-value	Indicates	Reject H_0?
<α	Strong evidence against the null hypothesis	Yes
>α	Weak evidence against the null hypothesis	No
=α	Marginal	NA

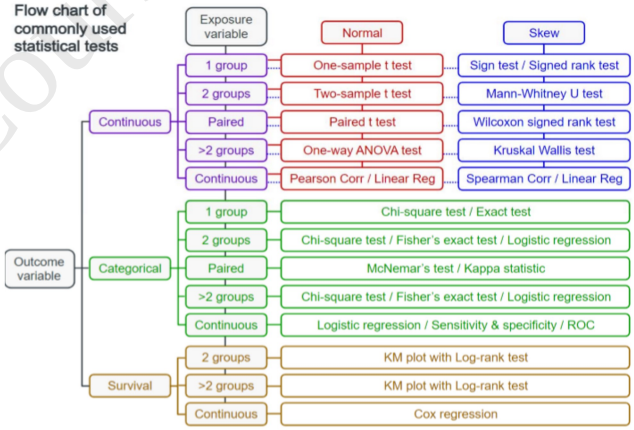
Increase power of significance test - larger sample - more reliable statistice, less likely to have type 1 (reject h0 when true) and type2 (retain h0 when false) errors.

Confidence Interval is a range [Lower bound, upper confidence bound] 95% CI says ' we are 95% confident that true values lie between LB and UB.

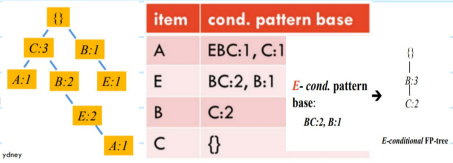
Association Rule Mining - Predict occurrence of an item based on other items in transaction for eg. (k-itemset)
Support count (sigma) - itemset frequency.
Support(s) - normalised eg. Support count/total transcts.
Frequent itemset if s >= min_support (let this be 50% for eg.)
Association Rule - Itemset [X] -> Itemset [Y] & confidence (c) is how often [Y] occurs with X so - [X][Y]/[X]
So generate frequent item sets and high confidence rules from each frequent itemset.

There are 2^d itemsets (eg. 2^5 for A,B,C,D,E)
To narrow down, Apriori Principle -> If an itemset is frequent, all its subsets are also frequent. If itemset is infrequent, all its supersets are also infrequent. Eg. If AB infreq, sets below with AB can be pruned.
FP - Growth - Highly compacted database structure, avoid costly repeated database scans. Avoids candidate generation. Construction: 1st scan: find frequent items (single) and order into list L with descending frequency. 2nd scan: Order freq. items in each transaction according to list L and construct FP-tree by adding each freq. Ordered transaction to tree.

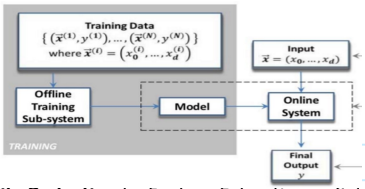
Flow chart of commonly used statistical tests



Apriori Algorithm - add each item to the initial list of candidate itemsets. sort and return initial 1 - item sets, calc. Support count and prune itemsets that dont meet support threshold, generate the next list of itemset (maybe 2 item sets), prune again, generate next list of itemsets, prune etc. From frequent itemset(s), now make all possible association rules. Calc, confidence of rules and return consequences that pass the conf. Threshold. Add these rules to the rule accumulator. Recursively evaluate rules by Generating candidate consequence itemsets, updating rules, returning consequences that pass the confidence threshold and then recursing with new consequence candidates. Likewise mine all assoiation rules. Bottleneck of Apriori - generating huge candidate sets, multiple scans of database.



Conditional F-P Tree:
1: Conditional Pattern Base
2: Make FP-tree for each Item
3: Find all freq item sets for each Item. Eg, for E, E:3, EB:3, EC:2, EBC:2. (Do for A, E,B, C)



Dimensionality Reduction by Feature Extraction or Selection (Filter, Wrapper)
Principal component analysis(PCA) - extract dimension(s) that captures most of the variation. Useful when dataset is highly correlated and has alot of dimensions and you want a subset that captures most of the variation. PCs are dimensions along which data is most spread out. 1st pc captures greatest variance, 2nd pc the second greatest variance etc.
Correlation shows there is redundancy in data. So pcs is used to reduce original variables that still explain the variance in. Captured by covariance matrix.
Covariance - Sign is impt, + means dimensions Inc tgt. - means one inc, the other dec. 0 means the 2 dimensions are independent
1. standardize data, 2. Calc Covariance matrix, 3. Find eigenvalues and eigen vectors of cov. matrix, 4. plot eigenvectors/PCs over scaled data.
Advantages - Visualization (using the best variables provide better insights) (hidden categories or clusters can be identified) Eg. do PCA on dataset and choose best PCs (representing most of the var) to reduce data to (2) dimensional spaces.

Hierarchical Clustering - A set of nested clusters organized as hierarchical tree
- Agglomerative (bottom up)(each point starts in own cluster. Find closest pair of clusters(min dist. Bet pts) merge. Find new distance between clusters.
- Divisive (all pts start in one cluster, split recursively down the hierarchy.

Partitional Clustering - division of data objects in non-overlapping subsets/clusters so each **K-means Algorithm** object is in only one subset.
- (1. partition objects into k nonempty subsets/clusters. 2. compute centroids(mean point) of the clusters as seed points. 3. Assign each object to its nearest seed point. Update new seed points by repeating step 2. Repeat till assignment does not change) or when k must be specified, (1. select k points as initial centroids. 2.Form clusters by assigning points to nearest centroid. 3. Recompute centroids. Repeat until centroids don't change) -> O(n*k*i*d)
N = no. of points, k = no. of clusters, i = no. of iterations, d = no. of attributes

Cluster Validity - External Index: checking cluster label against external labels
Homogenity (0-1) - checking for part of a single class. Completeness(0-1) - checking for part of a single cluster. V-measure - harmonic mean of both.
Internal Index: Checking goodness of the clustering without external:
Sum of Squared Error (SSE) - sum of dist to nearest means = $i\sum K\ 1\sum x\ dist^2(c,x)$
Silhouette Coefficient - For individual point i, calc A(avg dist or i to points in cluster). calc B(avg dist of i to points in next cluster)
silhouette coefficient, s = 1-A/B if A<B oe s = B/A -1 if A>=B

Choosing k - High avg silhouette indicate points far away from neighbour clusters. When uniform cluster silhouettes are close to avg coefficient in bar chart, means similar quantity of clusters as tested.
Pre-processing for clustering - Data Cleansing, Data Transformation, Data Normalisation, Dimensionality Reduction.

Supervised ML - Simple, Multiple Linear Regression & Logistic Regression
Predictive Modelling: Research to understand past attempts to solve problem -> Data Collection such as timeframe needed and transform raw data for efficiency-> Model Selection by building and testing several models. Split into training and test set. Pick best performer and train model to select optimized parameters -> Results-analyse model for test accuracy, and if additional data is needed for better accuracy and make adjustments as needed. Compare with other attempts.
Def. Obj. -> Collect data->prep data->develop models->train model-> analyze/evaluate ->publish->monitor/operate->def.obj

Goal: build models that generalise to unseen data in regression, classification and anomaly detection.
When training on many sample, should have low bias and low variance. High variance means, different training sets have differing results (overfitting). High bias means model never fits training sets well and makes mistakes so underfitting.

Simple Linear Regression - Method to find the best fit line between dependent variable Y (response) and one independent variable X (predictor).$Y = \alpha + \beta x + e$
e=sum of sqrd errors=$\sum(Ytrue - Ypred)^2$
Coefficient of determination (R^2) = $1 - \frac{(\sum ytrue - ypred)^2}{(\sum ytrue - yfit)^2} = 1 - \frac{SSE}{SST}$
0-1 -> higher is better, shows goodness of fit but for precision: standard error(S) = $sqrt\left(\frac{SSE}{N}\right)$
Model Acceptance testing with S: should fall within prediction interval which is generally : 95% of predictions should fall within Ypred +/- 2 x S eg. If predictions must be within 5k, then s<=2.5k

Multiple Linear Regression: explain relationship bet. Two or more explanatory variables, one response variable.
Residual Plot: (will have green training points and purple predicted points) considered good fit if symmetrically distributed and cluster towards center, there are no clear patterns and cluster around y=0. Bad if y axis is off balance (points towards one side), inconsistent variance, non-linear and outlier(only one part of the horizontal line)

classification vs regression. Classification assigns a class vs assigning a numerical value. Output is discrete/categorical variable vs continuous variable. (colour prediction vs house price)
Logistic regression: predict prob. of categorical label**, eg.** probability of defaulting on a loan given amount of debt and late payment count. Applying linear regression for classification is not useful. Logistic or sigmoid function used. When classifying, use threshold y>=0.5 predict y= 1, y<0.5, predict y=0.

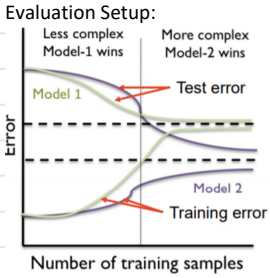
Large feature spaces in plots create overfitting problems. Prevent by using **regularisation** which adds penalty that increases as coefficients(β) get larger. More weight given to error term, more large coefficients are discouraged. To select parameters like penalty and reg. Strength, exhaustive search thu combinations of specified values can be used. Perform n-fold cross validation for each combination.

Decision Trees: tree is constructed in top-down recursive divide and conquer manner. Training examples are at root. Examples partitioned recursively based on selected attributes.

Measuring entropy: higher entropy means higher uncertainty, vice versa
Y=yes Y=no
- $H(Y) = -(0.5*\log_2(0.5) + 0.5*\log_2(0.5)) = 1$
- $H(X) = -(0.5*\log_2(0.5) + 0.25*\log_2(0.25) + 0.25*\log_2(0.25)) = 1.5$

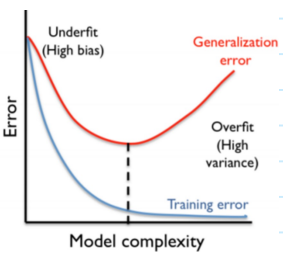
y=math y=history y=cs
Specific Conditional entropy: out of 4 math, 2 yesses and 2 no, out of history both no, out of cs both yes
$H(Y|X=Math) = -(0.5*\log_2(0.5) + 0.5*\log_2(0.5)) = 1$
$H(Y|X=History) = -(0 + \log_2(1)) = 0 + 0 = 0$
$H(Y|X=CS) = 0$
Conditional entropy: sum of profuct of H(xindividual) and H(Y/X)
$H(Y|X) = 0.5*1+0.25*0+0.25*0 = 0.5$
Information Gain is change in entropy after making a decision in the tree
IG(Y/X) = H(Y) - H(Y/X)

Evaluation Setup:



- Test error decreases
- Training error increases
- Two converge to asymptote
- If we can get more data, model 2 eventually wins
- Neither model will improve much with more data than we already have

- The dashed line on right shows point where we switch form underfitting to overfitting
- Goal: Find this dotted line
- Generalisation error should model application as closely and reliably as possible
  - Sample must be representative
  - Larger sample better



Data Drift: Typical Train test setups assume stationary. (mean, variance etc dont change over time. Should be near true for train,test samples butonly near true in production for a short while.
Solution: Monitor offline metric on live data, if large changes, retrain on new data, online/incremental learning.