XML vs HTML:

* HTML is rendered in a browser and made readable to humans. Used to describe presentation or formatting
* XML is used to make data readable to both humans and machine. Used to describe content and elements
* XML allows for complex schema definitions and descriptions of tags

XML vs JSON:

* Both are used to make data readable to both humans and machine. Used to describe content and elements
* XML has to be parsed with a dedicated parser
* JSON can be parsed directly by JavaScript and is much quicker to read and write
* JSON designed for speed and efficiency and used widely in noSQL databases

Missing Values:

* Values missing at random are caused unrelated to any another measured values
  + Test results but person X never took it and has no result
* Values missing not at random are caused related to some measured value of the data
  + Test results but person Y did poorly and did not want to disclose

Imputations:

* Deletion:
  + Easier to analyse but may produce bias for small sample size
* Manual Correction:
  + Time consuming and is it possible
* Zero Fill:
  + Won’t break any application but limited utility for analysis
* Mean Fill:
  + Good for supervised classification
  + Reduces variance and may make the distribution biased
  + Relationships to other features change
* Median Fill:
  + Better estimator depending on distribution

Outliers:

* Global outliers / Point anomaly deviates significantly from the dataset for no apparent reason
* Contextual outliers / Conditional Outliers deviate significantly from the dataset due to some reason
  + The temperature is 5 degrees outside – but is it in Winter or Summer?

Quantiles:

* Use Type 6 quantiles where
* Inner fences are at and at
* Outer fences at from respective quantiles

Collaborative Filtering:

* Method 1:
  + Impute missing values with mean
  + Similarity is equal to the Euclidian Squared Distance
* Method 2:
  + Calculate the Euclidian Squared Distance for non-missing paired samples
  + Scale the result by
* Correlation:
  + Similarity is the reciprocal of the Euclidian Squared Distance (method doesn’t matter)
  + Choose a value of k for the number of similar items

Parallel Coordinates

* The ordering of attributes is important to see visual correlation
* Easier to see if values are scaled between 0 to 1
* Good method of plotting high dimensional data

k-Means:

* Select k seed points
* Assign each object to the cluster with the nearest seed point
* Repeat until it doesn’t change
* Centroids of clusters is the average of all points in the cluster

Dissimilarity Matrix:

* Consists of distances between clusters
* Better to order and visualise using VAT

VAT:

* Visually suggests number of clusters and features allocated to it
* Dark blocks ill appear when tightly correlated groups exist in the data

Minimal Single Linkage Clustering (Agglomerative):

* Connects the closest objects of each cluster
* Similar to a bottom-up Merge Sort
* Minimum complete linkage tends to break large clusters up but is susceptible to noise and outliers
* Max complete linkage is less susceptible to noise and outliers but may fail to break clusters up

Binning:

* Equal width has equal length intervals
* Equal frequency has equal frequencies in each bin

KNN:

* Compute distances from the given value to the established labelled values
* Use the k nearest neighbours (shortest distance from values) to determine the label
* If k is small, then it is sensitive to noise but if k is too large it may include points from other classes

DT:

* Make a contingency table
* Calculate entropy of the class label
* Calculate entropy for each feature
* Information Gain is entropy of label minus entropy of feature
* If the data is continuous, discretise them into categories and split by entropy

Blocking:

* Worst Case is
* Average Case is
* Best Case is
* More blocks mean faster computation but less accuracy whilst less blocks results in slower computation but a higher accuracy

Party Protocols:

* Two Party:
  + Use a one-way hash function on the values
  + Share hashed values with other party
  + Prone to dictionary attacks – Generate a dictionary of hashed values until a matching one is found
* Three Party:
  + Eliminates possibility of a dictionary attack from party B
  + Party A and B sends hashed values to party C which checks the matches for them
  + Prone to dictionary and frequency attacks from C
  + Can be prevented by adding in a salt before hashing and adding random values

Bloom Filters:

* where h is number of matching bits and b is the total number of 1’s in each bloom filter
* If a single index of the hashed string is a 0 then it is definitely not part of the bloom filter
* Be aware of false positives which can be caused coincidentally
* Sparse bloom filters mean less privacy but faster matching whilst full bloom filters results in more privacy but prone to false positives

Blockchain:

* Less administration, faster transactions, ability to be anonymous and may be more secure
* Ledger is a sequence of blocks with each containing a header and some fact
* Block ID is derived from the ID of its parent block
* Header / Block ID contains parent block ID, timestamp and fact

Chaining characteristics:

* If a block ID changes then all subsequent block IDs require recalculation
* Not viable since blockchain is a P2P network and requires a node to convince each block to recalculate

Cryptography:

* Public key + Original Message -> Encryption Algorithm -> Encrypted Message
* Private key + Encrypted Message -> Decoding Algorithm -> Original Message

Digital Signatures:

* Sign content with a private key to make it digitally signed
* Can check if contents have changed by using the public key on the digital signature and the hash function on the document contents
* If the resulting hashes are equal, then there is no modification
* If the resulting hashes are not equal, then there has been a modification to the content

k-anonymity:

* Explicit Identifier is a PK
* Quasi Identifier is a combination of non-sensitive attributes
* Sensitive Attributes are information that requires to be private
* Guarantees that the individuals of the data cannot be reidentified whilst the data remains useful for research and analysis
* A dataframe is k-anonymous if every record in the table is indistinguishable from at least k-1 other records (with respect to quasi identifiers)
* Can generalise by being less specific or more broad
* Can supress by limiting values displayed with \*

Attacks on k-anonymity and Ɩ-diversity:

* Homogeneity:
  + Possible when k-anonymity lacks diversity in the sensitive attribute
  + Example: If person X knows that person Y is A years old from C city, then person X can narrow down person Y to a quasi-identifier. If the sensitive information S is all equal for that quasi identifier, then person X can conclude person Y has sensitive information S
* Background:
  + Possible when person X knows background information on person Y
  + Example: If person X knows person Y comes from city C which has a low chance of disease D occurring, then person X can conclude that person Y is more likely to have disease D’ given they can narrow person Y down to a set of quasi identifiers
* Solution:
  + Use Ɩ-diversity which aims to make the sensitive attributes diverse
  + At least Ɩ groups of sensitive attributes per quasi identifier
* To reduce risks of reidentification, make it k-anonymous and Ɩ-diverse

Location and Trajectory Privacy (Inference attack):

* Cloaking:
  + K-anonymity – makes individuals k-anonymous
  + Spatial cloaking – adapts spatial precision based on number of people in the area
  + Temporal cloaking – reduces frequency of information available
* Obfuscation:
  + Masks an individual’s location to an approximate area
  + Degrades quality of information and works under the assumption that the more obfuscation results in more privacy
* k-anonymity will yield higher privacy but much slower wait time whilst obfuscation will be instant with no wait time but have with less privacy and accuracy

Differential Privacy:

* Promises that the chance of the noisy result being equal to R is approximately the same regardless if an individual participates or not
* A is the probability of participating and B is the probability of not participating where you want to be able to guarantee that A is approximately B
* Global Sensitivity:
  + Maximum difference when adding or removing any individual from the dataset
  + Sum of maximum differences if we use multiple queries
* Privacy Loss Budget:
  + Choose a value k such that the presence or absence of any individual from the dataset will not have a considerable effect on the result
  + Want a value such that
  + Smaller the k, smaller the privacy loss but smaller utility for research and analysis
* Guarantees that the presence or absence of a user cannot be revealed after releasing the results but does not prevent attackers from drawing conclusions about individuals
  + Example: If 220 females and 200 males took subject S, the attacker can conclude that there were indeed 220 females and 200 males, but not if a specific person took the subject S