① Noise ② Outliers

A_1	A_2	A20	
7.85	Good	Avg	Y
6	Good	Bad	N
5.1	Very bad	Bad	Y

Senex in corona

Throw out outliers! (Out of scope)

---

f_1 f_2 --- f_{15} ... f_{50} TARGET(Y/N)

X
NA

Missing values :- (i) Drop the missing records
(ii) If too much data is missing
one fix: Eg: If out of 10k, 8k have values
, take average of feature

Another fix:

Lets say out of 10k, 6k are positive examples and out of 6k, 5.8k have values, 200 will get average of 6k

repeat for all classes (Eg: positive/negative)
1, 2, 3, ... 10

→ Another way: f_1 f_2 ---- f_{15} ---- f_{50} Target Y/N (1/0)

D_1
D_2
D_3
D_4

find nearest neighbors of this point in $\mathbb{R}^{50}$

· find 5 NN and check their feature if missing, if not, take average!

Duplicate data

diff sources may have same data

→ the data point may skew the data and prejudice the learning, if it is repeated → misleading prediction by kind of saying more data

(Sometimes good tho, when less features or during random oversampling to clear bias due to dataset)

Similarity & dissimilarity measures

eg: Distance : Euclidean

Similarity and Dissimilarity Measures

- Similarity measure
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Dissimilarity measure
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

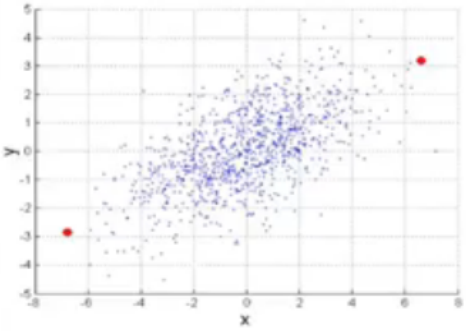
Where r is a parameter, n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects $\mathbf{x}$ and $\mathbf{y}$.

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
  - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors
- $r = 2$. Euclidean distance
- $r \to \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component of the vectors
- Do not confuse r with n, i.e., all these distances are defined for all numbers of dimensions.

Mahalanobis Distance

$$\text{mahalanobis}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})$$



Σ is the covariance matrix

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.

many distance metrics from previous class

Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$ for all x and y and $d(\mathbf{x}, \mathbf{y}) = 0$ only if $\mathbf{x} = \mathbf{y}$. (Positive definiteness)
2. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$. (Symmetry)
3. $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ for all points $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$. (Triangle Inequality)

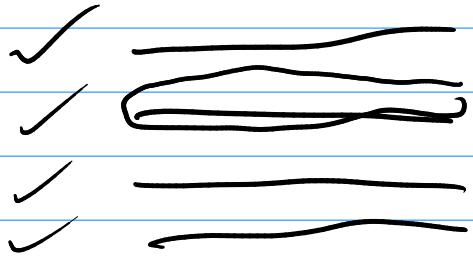
where $d(\mathbf{x}, \mathbf{y})$ is the distance (dissimilarity) between points (data objects), $\mathbf{x}$ and $\mathbf{y}$.

- A distance that satisfies these properties is a metric

Machine Learning: feature engineering

eg. Agg height & weight into BMI for finding diseases

Sampling (if too many points)



→ Sampling with replacement
w/o replacement

Data Preprocessing

- Aggregation
- Sampling
- Dimensionality Reduction
- Feature subset selection
- Feature creation
- Discretization and Binarization
- Attribute Transformation

→ get all data in a txn database for a month of January

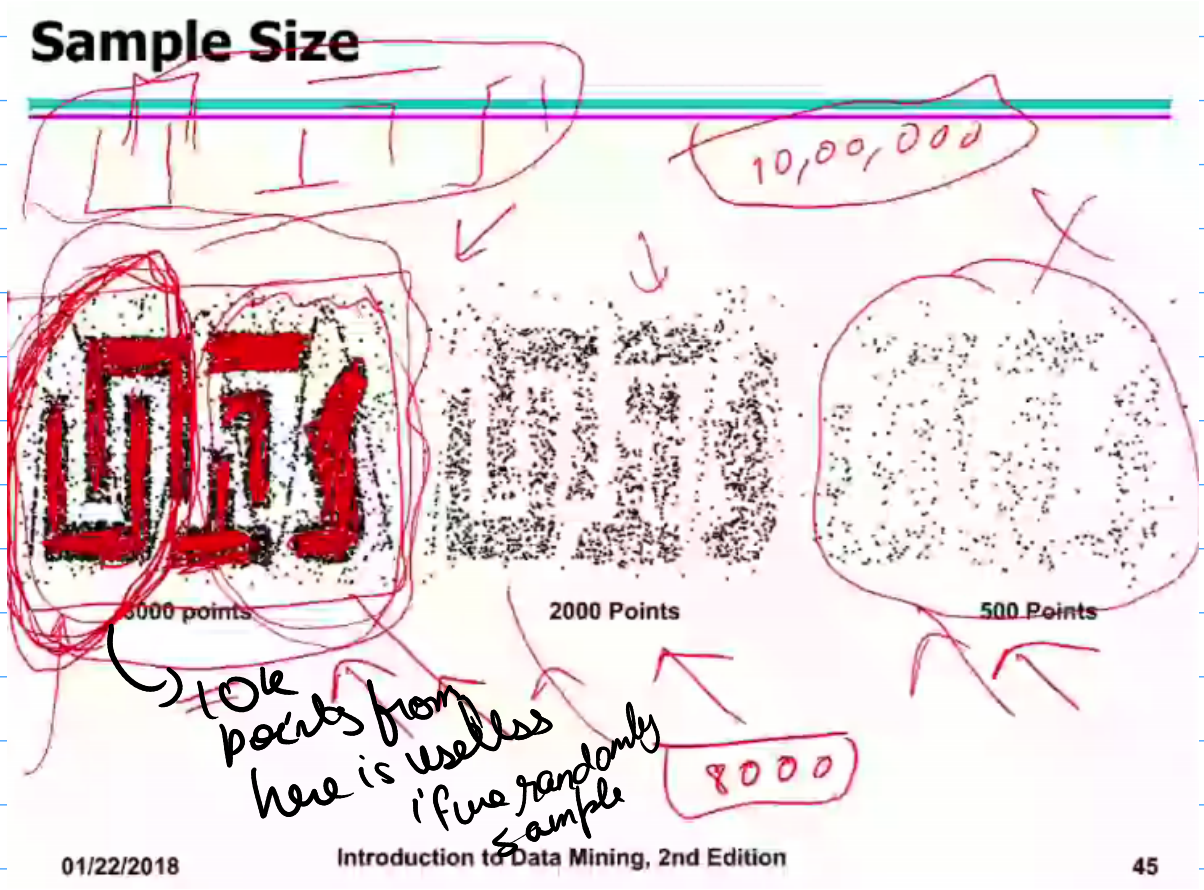
(Aggregation) like concatenation of datapoints

Another way of aggregation

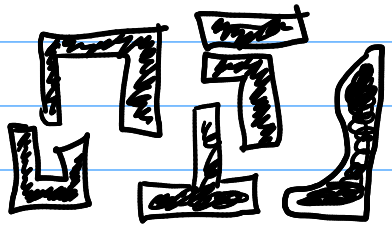
⇒ involves algebra of features

eg: BMI

We also need to note that while sampling points, we can't sample from a single place



Some points from here is useless if we randomly sample



So we do something called stratified sampling

Stratified sampling is all about dividing your space into partitions and sample from each (maybe equally ??)

You may not know this, you need to have a reasonable understanding of the data beforehand for stratified sampling

But you may be unlucky then as well since you might sample a part of the partition (But if can we do there)

Another thing we should note: PCA

→ Lets say we have 1000 features, and as data scientists, we don't know what feature is useful and what feature is useless (eg: in maybe hiresolution images or sth where the features aren't that very apparent)

* We need an algorithm to reduce the number of features.