EXTERNAL CLUSTER VALIDATION METRICS

1. RAND INDEX

True positive (TP)
False positive (FP)
True negative (TN)
False negative (FN)

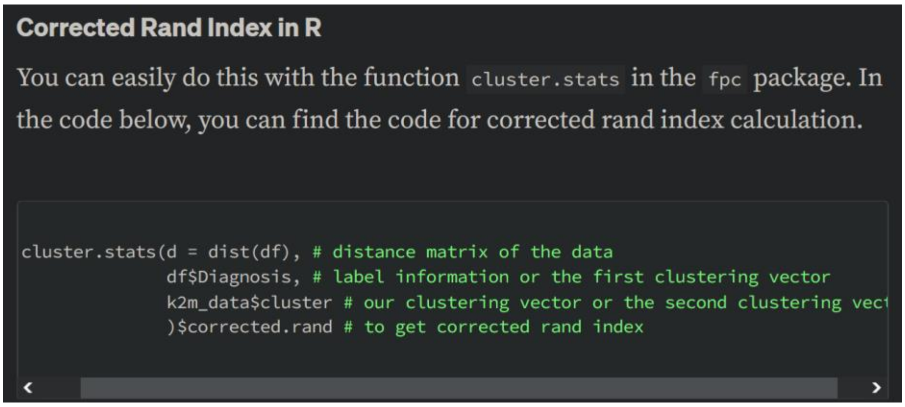
$$\text{Rand Index} = \frac{TP+TN}{TP+FP+TN+FN}$$

1 means the two clustering outcomes match identically

R docs: https://www.rdocumentation.org/packages/fossil/versions/0.4.0/topics/rand.index

2. AJUSTED RAND INDEX

È più robusto: l'ARI offre una misura normalizzata che tiene conto della possibilità che le assegnazioni possano coincidere per caso.



Corrected Rand Index in R

You can easily do this with the function `cluster.stats` in the `fpc` package. In the code below, you can find the code for corrected rand index calculation.

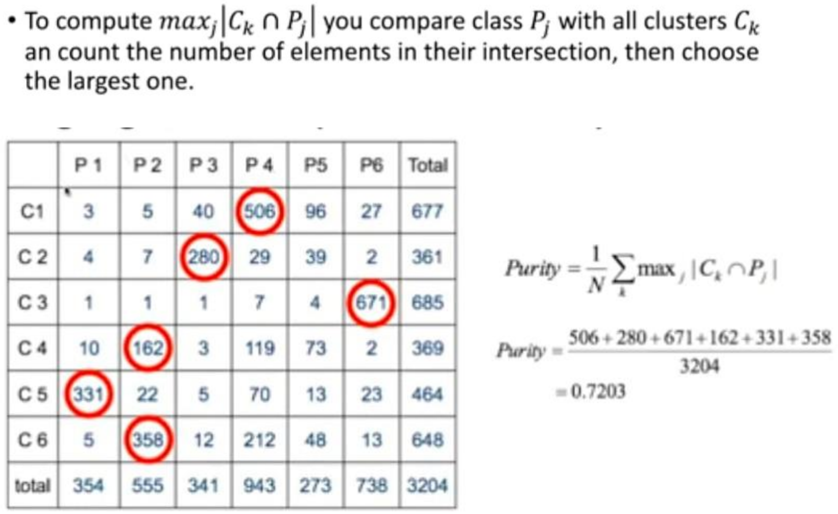
```
cluster.stats(d = dist(df), # distance matrix of the data
              df$Diagnosis, # label information or the first clustering vector
              k2m_data$cluster # our clustering vector or the second clustering vect
              )$corrected.rand # to get corrected rand index
```

Or R docs: https://www.rdocumentation.org/packages/mclust/versions/6.1/topics/adjustedRandIndex

3. PURITY

- Represents the fraction of correctly classified data points when assigning each cluster to the most frequent ground truth label in that cluster.
- **Range**: 0 to 1 (1 means perfect match).

$$Purity = \frac{1}{N}\sum_{k} max_j|C_k \cap P_j|$$



- To compute $max_j|C_k \cap P_j|$ you compare class P_j with all clusters C_k an count the number of elements in their intersection, then choose the largest one.

	P 1	P 2	P 3	P 4	P5	P6	Total
C1	3	5	40	506	96	27	677
C 2	4	7	280	29	39	2	361
C 3	1	1	1	7	4	671	685
C 4	10	162	3	119	73	2	369
C 5	331	22	5	70	13	23	464
C 6	5	358	12	212	48	13	648
total	354	555	341	943	273	738	3204

$$Purity = \frac{1}{N}\sum_{k}\max_j |C_k \cap P_j|$$

$$Purity = \frac{506+280+671+162+331+358}{3204}$$

$$= 0.7203$$

k= numero di cluster

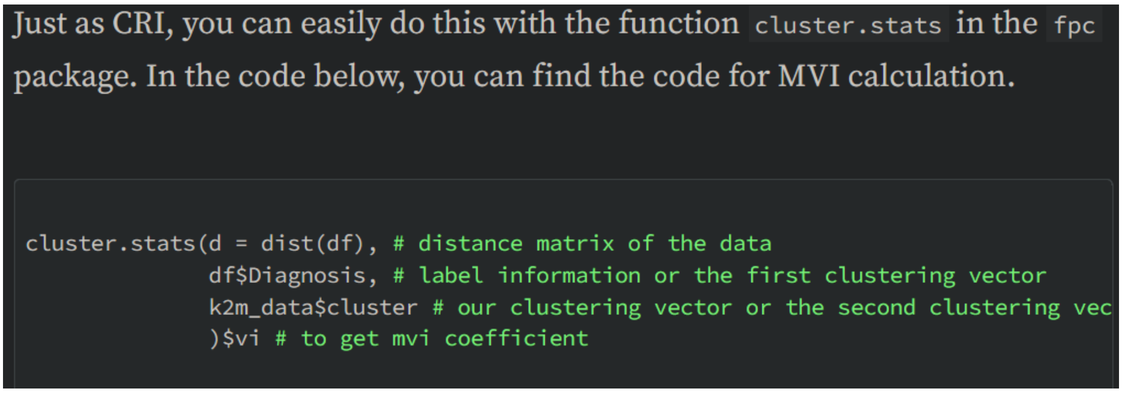
R docs: https://search.r-project.org/CRAN/refmans/funtimes/html/purity.html

4. MVI

Meila’s variation of information (MVI) is a measure used to assess the similarity between two different clustering solutions for a given dataset. It is based on the idea that **the similarity between two clustering solutions can be measured by the amount of information that is gained or lost when going from one clustering solution to the other**.

MVI compares the **entropy** of each clustering solution, and ranges from **0 to log(n)** where n is the number of observations.

A lower MVI value indicates that the two clustering solutions are more similar.



Just as CRI, you can easily do this with the function `cluster.stats` in the `fpc` package. In the code below, you can find the code for MVI calculation.

```
cluster.stats(d = dist(df), # distance matrix of the data
              df$Diagnosis, # label information or the first clustering vector
              k2m_data$cluster # our clustering vector or the second clustering vec
              )$vi # to get mvi coefficient
```

Rdocs : https://www.rdocumentation.org/packages/fpc/versions/2.2-13/topics/cluster.stats

5. NORMALIZED MUTUAL INFORMATION

- The mutual information (MI) between U and V is calculated by:

$$\mathrm{MI}(U,V)=\sum_{i=1}^{|U|}\sum_{j=1}^{|V|}P(i,j)\log\left(\frac{P(i,j)}{P(i)P'(j)}\right)$$

where $P(i,j)=|U_i\cap V_j|/N$ is the probability that an object picked at random falls into both classes U_i and V

- The normalized mutual information is defined as

$$\mathrm{NMI}(U,V)=\frac{\mathrm{MI}(U,V)}{\mathrm{mean}(H(U),H(V))}$$

Normalized Mutual Information (NMI) is a normalization of the Mutual Information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation).
This measure is not adjusted for chance. Therefore adjusted_mutual_information might be preferred.

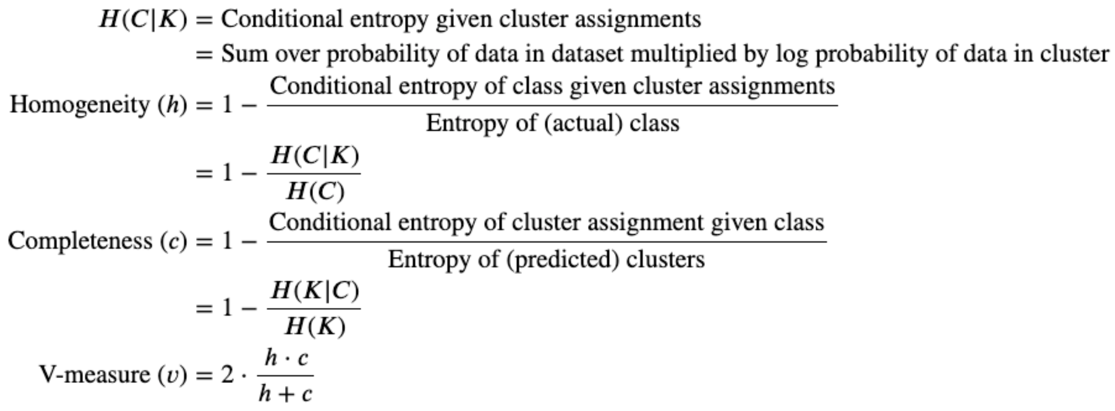
$$H(U)=-\sum_{i=1}^{|U|}P(i)\log(P(i))$$

$H(U)$ → entropy

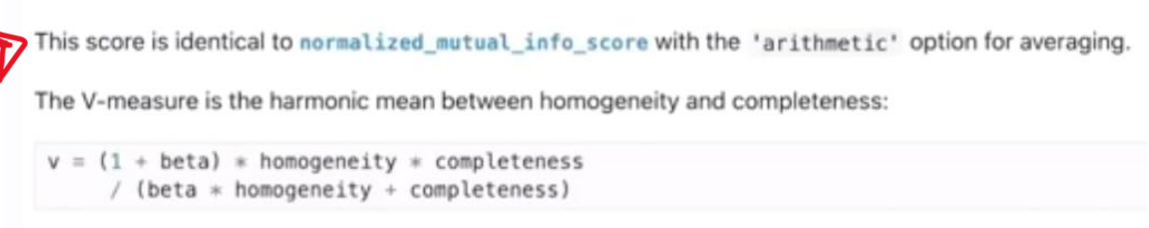
R docs: https://search.r-project.org/CRAN/refmans/aricode/html/NMI.html

6. V-MEASURE (HOMOGENEITY & COMPLETENESS)

- Combines homogeneity (same class members are clustered together) and completeness (all class members appear in the same cluster).



$$\begin{aligned}
H(C|K) &= \text{Conditional entropy given cluster assignments} \\
&= \text{Sum over probability of data in dataset multiplied by log probability of data in cluster} \\
\text{Homogeneity } (h) &= 1 - \frac{\text{Conditional entropy of class given cluster assignments}}{\text{Entropy of (actual) class}} \\
&= 1 - \frac{H(C|K)}{H(C)} \\
\text{Completeness } (c) &= 1 - \frac{\text{Conditional entropy of cluster assignment given class}}{\text{Entropy of (predicted) clusters}} \\
&= 1 - \frac{H(K|C)}{H(K)} \\
\text{V-measure } (v) &= 2 \cdot \frac{h \cdot c}{h + c}
\end{aligned}$$



This score is identical to `normalized_mutual_info_score` with the `'arithmetic'` option for averaging.

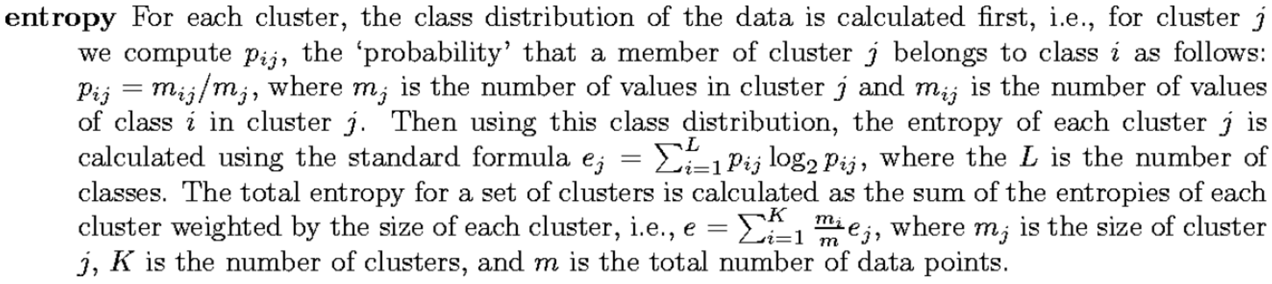
The V-measure is the harmonic mean between homogeneity and completeness:

```
v = (1 + beta) * homogeneity * completeness
     / (beta * homogeneity + completeness)
```

R docs: https://search.r-project.org/CRAN/refmans/clevr/html/v_measure.html

ENTROPY

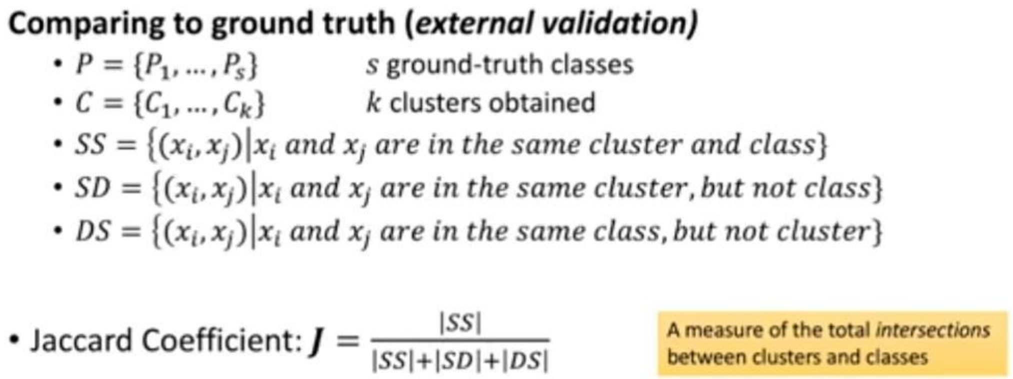
- **M**easures the "disorder" or randomness in cluster assignments relative to the ground truth clusters. Lower entropy indicates better clustering.
- **Range**: 0 (perfect alignment) to higher values as disorder increases.



entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

7. JACCARD COEFFICIENT (/JACCARD SIMILARITY)

- Measures similarity between predicted and ground truth clusters by counting common elements.

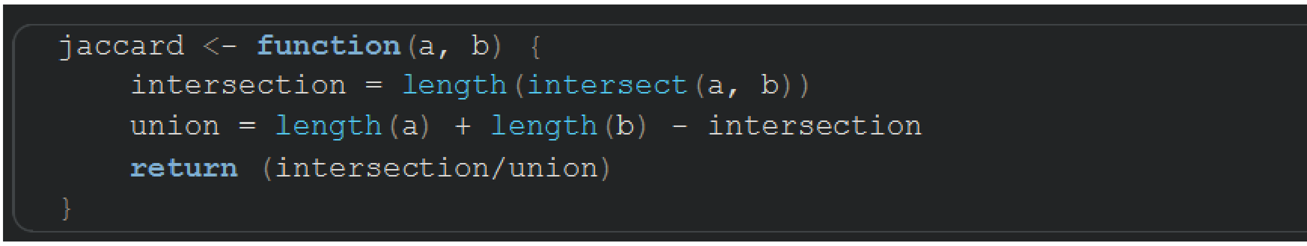


Comparing to ground truth (*external validation)*

- $P = \{P_1, \dots, P_s\}$ s ground-truth classes
- $C = \{C_1, \dots, C_k\}$ k clusters obtained
- $SS = \{(x_i, x_j) | x_i \text{ and } x_j \text{ are in the same cluster and class}\}$
- $SD = \{(x_i, x_j) | x_i \text{ and } x_j \text{ are in the same cluster, but not class}\}$
- $DS = \{(x_i, x_j) | x_i \text{ and } x_j \text{ are in the same class, but not cluster}\}$

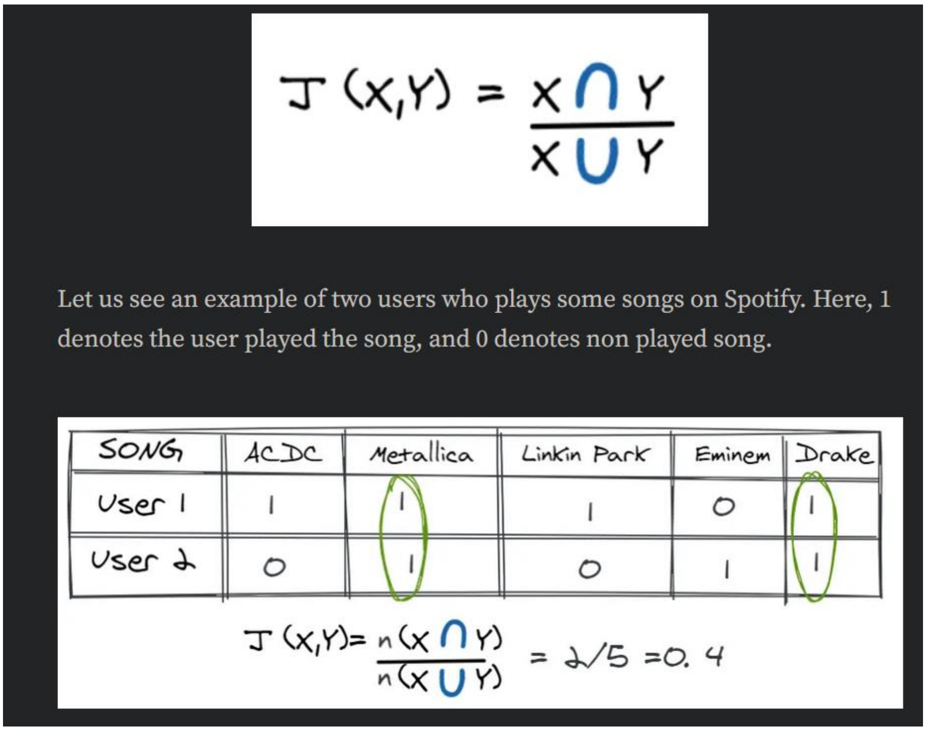
- Jaccard Coefficient: $J = \frac{|SS|}{|SS|+|SD|+|DS|}$

A measure of the total *intersections* between clusters and classes



```
jaccard <- function(a, b) {
    intersection = length(intersect(a, b))
    union = length(a) + length(b) - intersection
    return (intersection/union)
}
```

Dettagli: https://www.r-bloggers.com/2021/11/how-to-calculate-jaccard-similarity-in-r/



$$J(X,Y) = \frac{X \cap Y}{X \cup Y}$$

Let us see an example of two users who plays some songs on Spotify. Here, 1 denotes the user played the song, and 0 denotes non played song.

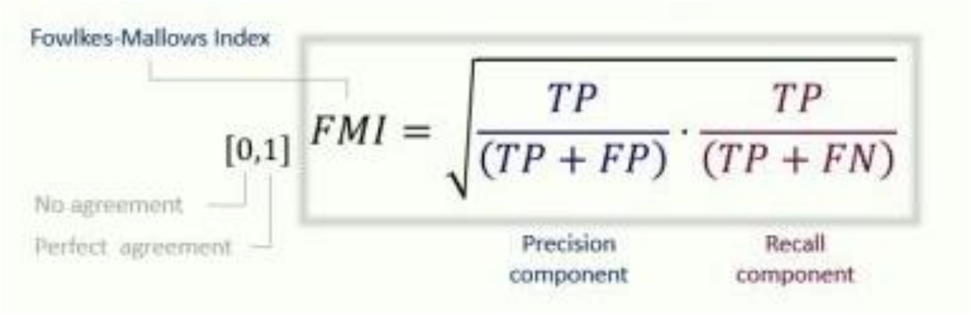
SONG	AC DC	Metallica	Linkin Park	Eminem	Drake
User 1	1	1	1	0	1
User 2	0	1	0	1	1

$$J(X,Y) = \frac{n(X \cap Y)}{n(X \cup Y)} = 2/5 = 0.4$$

R docs: https://search.r-project.org/CRAN/refmans/mlr3measures/html/jaccard.html

8. FOWLKES-MALLOWS INDEX

- The *Fowlkes–Mallows index* is an external evaluation method that is used to determine the similarity between two clusterings
- **Range**: 0 to 1



$$FMI = \sqrt{\frac{TP}{(TP + FP)} \cdot \frac{TP}{(TP + FN)}}$$

R docs: https://search.r-project.org/CRAN/refmans/dendextend/html/FM_index.html