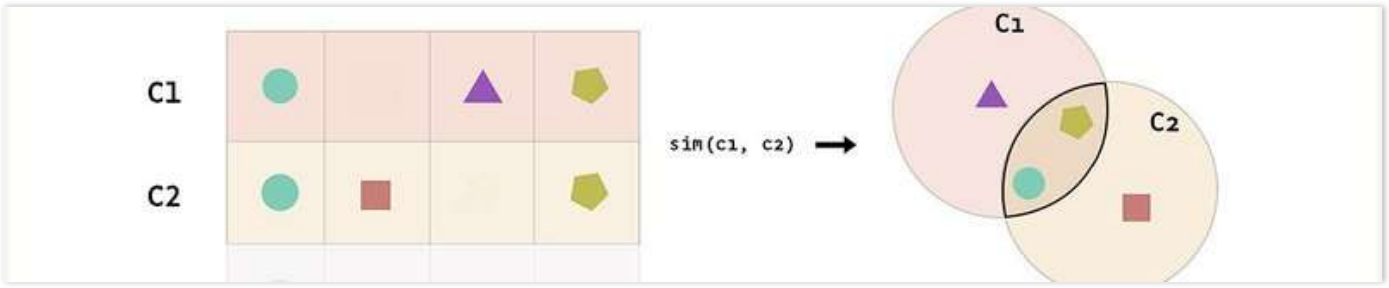
You are reading **Glossary**



Author: Fatih Karabiber
Ph.D. in Computer Engineering, Data Scientist

Jaccard Similarity

Contents Index +

You should already know:

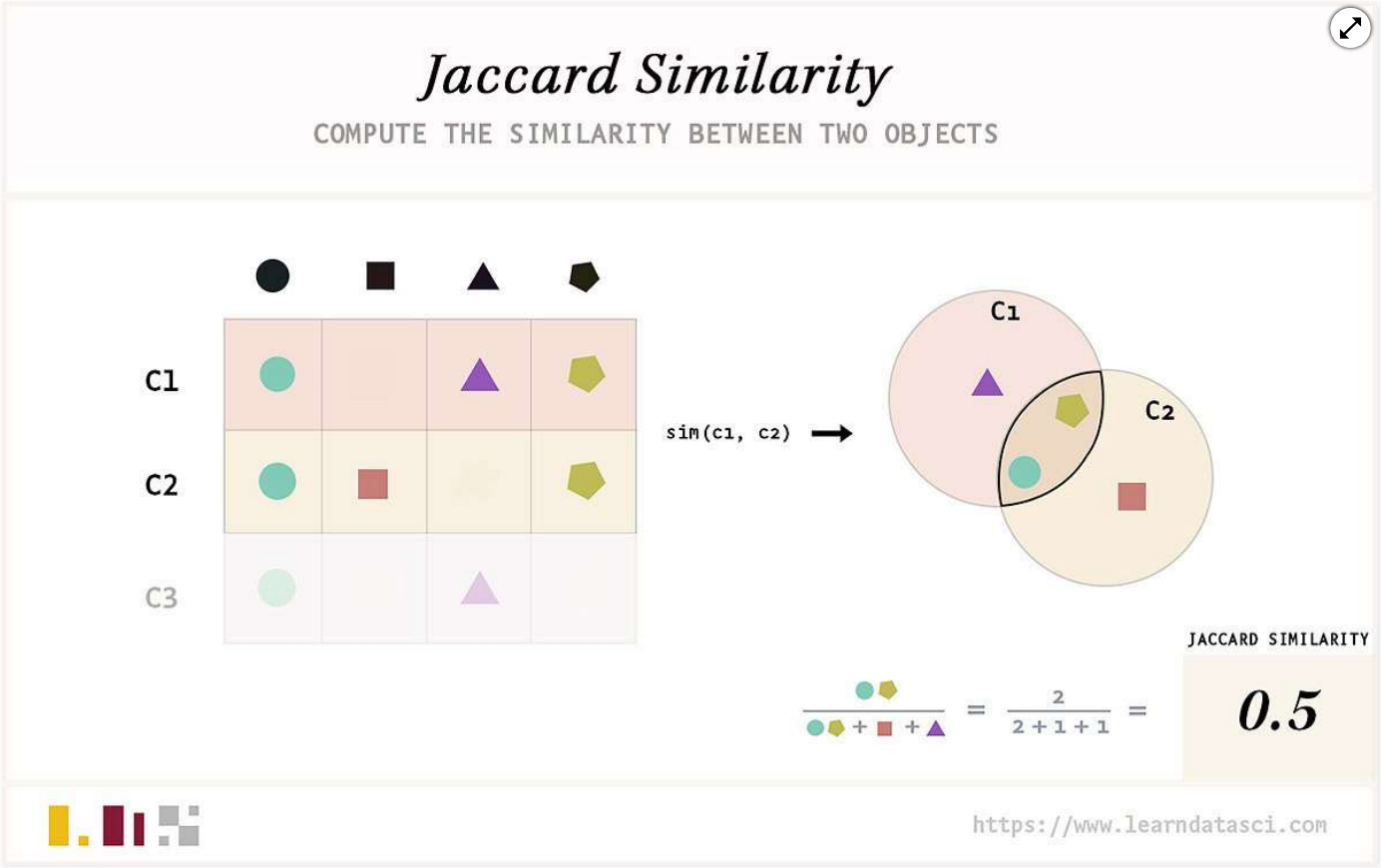
Basic Python — Learn Python and Data Science concepts interactively on Dataquest.

What is Jaccard Similarity?

Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents. Jaccard similarity can be used to find the similarity between two asymmetric binary vectors or to find the similarity between two sets. In literature, Jaccard similarity, symbolized by J, can also be referred to as **Jaccard Index**, **Jaccard Coefficient**, **Jaccard Dissimilarity**, and **Jaccard Distance**.

Jaccard Similarity is frequently used in data science applications. Example use cases for Jaccard Simila

- **Text mining:** find the simil
  number of terms used in t
- **E-Commerce:** from a mark
  millions of items, find simila customers via their purchase history
- **Recommendation System:** Movie recommendation algorithms employ the Jaccard Coefficient to find similar customers if they rented or rated highly many of the same movies.



1. Jaccard Similarity for Two Binary Vectors

The Jaccard Similarity can be used to compute the similarity between two asymmetric binary variables. Suppose a binary variable has only one of two states: 0 and 1, where 0 means that the attribute is absent, and 1 means that it is present. While each state is equally valuable for symmetric binary attributes, the two states are not equally important in asymmetric binary variables.

1.1. Numerical Example

Say we are trying to comput
could have a binary attribute
store, where 1 indicates tha
that a product was not purc

Since there could be thousands of products in the store, the number of products **not** purchased by any customer is much higher than the number of products purchased. When computing the similarity between customers, we only want to factor in purchases of items. This means that the *item purchased* is an **asymmetric binary variable**, making a value of 1 more important than 0.

In the first step of a Jaccard Similarity measurement for two customers which consist of n binary attributes, the following four quantities (i.e., frequencies) are computed for the given binary data:

- a = the number of attributes that equal 1 for both objects i an j
- b = the number of attributes that equal 0 for object i but equal 1 for object j
- c = the number of attributes that equal 1 for object i but equal 0 for object j
- d = the number of attributes that equal 0 for both objects i and j.

Then, Jaccard Similarity for these attributes is calculated by the following equation:

$$J(i,j) = sim(i,j) = \frac{a}{a+b+c}$$

Notice the number of 0 matches is considered unimportant in this computation and is ignored because the items are asymmetric binary attributes.

Suppose that a customer transaction table contains 9 items and 3 customers. The similarity between objects is computed based only on the asymmetric attributes.

	item1	item2	item2	item4	item5	item6	item7	item8	item9
C1	0	1	0						
C2	0	0	1						
C3	1	1	0						

The similarity between each
by Jaccard Coefficient:

$$J(C1, C2) = \frac{a}{a+b+c} = \frac{1}{1+1+2} = 0.25 J(C1, C3) = \frac{a}{a+b+c} = \frac{}{2+}$$

These measurements suggest that $C1$ and $C3$ have similar shopping behavior, whereas $C2$ and $C3$ are not similar since they have purchased completely different items.

Lastly, instead of similarity, the dissimilarity or **Jaccard Distance** between two binary attributes can be calculated. The dissimilarity based on these attributes by the Jaccard Coefficient is computed as follows:

$$d(i, j) = \frac{b+c}{a+b+c} \implies 1 - sim(i, j)$$

1.2. Python Example

Below, a function is defined to compute Jaccard similarity between two binary vectors. You can also find this builtin to *scikit-learn*, under sklearn.metrics.jaccard_score.

```
import numpy as np

def jaccard_binary(x,y):
    """A function for finding
    intersection = np.logical
    union = np.logical_or(x,
    similarity = intersection
    return similarity

# Define some binary vectors
x = [0,1,0,0,0,1,0,0,1]
y = [0,0,1,0,0,0,0,0,1]
z = [1,1,0,0,0,1,0,0,0]

# Find similarity among the vectors
simxy = jaccard_binary(x,y)
simxz = jaccard_binary(x,z)
simyz = jaccard_binary(y,z)

print(' Similarity between x and y is', simxy, '\n Similarity between x and z is ', s
```


OUT:

```
Similarity between x and y is 0.25
 Similarity between x and z is  0.5
 Similarity between x and z is  0.0
```

2. Jaccard Similarity for Two Sets

The Jaccard similarity measures the similarity between two sets of data to see which members are shared and distinct. The Jaccard similarity is calculated by dividing the number of observations in both sets by the number of observations in either set. In other words, the Jaccard similarity can be computed as the size of the intersection divided by the size of the union of two sets. This can be written in set notation using intersection $(A \cap B)$ and unions $(A \cup B)$ of two sets:

$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$

where $|A \cap B|$ gives the nu
and $|A \cup B|$ gives the total
un-shared). The Jaccard Sim
values and 1 if the two sets
numerical values or strings.

Additionally, this function can be used to find the dissimilarity between two sets by calculating $d(A,B) = 1–J(A,B)$.

2.1. Numerical Example

Example: A simple example is given below to compute the Jaccard similarity between the following two sets.

- $A = \{0, 1, 2, 5, 6\}$
- $B = \{0, 2, 3, 4, 5, 7, 9\}$

Jaccard Similarity between two sets is calculated as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|\{0,2,5\}|}{|\{0,1,2,3,4,5,6,7,9\}|} = \frac{3}{9} = 0.33$$

2.2. Python Example

We can define a function to calculate the Jaccard Similarity between two sets of data in Python like so:

```
def jaccard_set(list1, list2)
    """Define Jaccard Similar
    intersection = len(list(s
    union = (len(list1) + len
    return float(intersection

# Define two sets
a = [0, 1, 2, 5, 6]
b = [0, 2, 3, 4, 5, 7, 9]

# Find Jaccard Similarity between the two sets
jaccard_set(a, b)
```


OUT:

```
0.3333333333333333
```

As we can see, the result is identical to the numerical example above. The two sets have a Jaccard Similarity of 0.33.

Get updates in your inbox

Join over 7,500 data science learners.

Enter your email

Subscribe

Meet the Authors

Ph.D. in Computer Engineering, Data Scientist

Associate Professor of Computer Engineering. Author/co-author of over 30 journal publications. Instructor of graduate/undergraduate courses. Supervisor of Graduate thesis. Consultant to IT Companies.

Back to blog index

Load Comments

Best Data Science Courses **Best Machine Learning Courses**

Best Udemy Courses

Data Science & Machine Learning Glossary **Free Data Science Books**

Privacy Policy