

Feature: Affinity

 For affinities to work on a site, the affinity feature must be enabled for the site. To enable the feature, if it's not already enabled, contact [Frosmo support](#).

 While this feature is officially known as "Super-Affinity", the documentation talks about "affinity" for simplicity.

Affinity (or **visitor affinity**) measures a visitor's level of engagement with the **items**, such as retail products or online games, on a website. Affinity indicates what an individual visitor is interested in. The more a visitor interacts with specific items, and the more interest they therefore show in those items and the attributes the items represent, the greater their affinity for the items and for similar items. You can use affinity to personalize the content that visitors interact with and the user journeys they experience based on their individual interests and preferences.

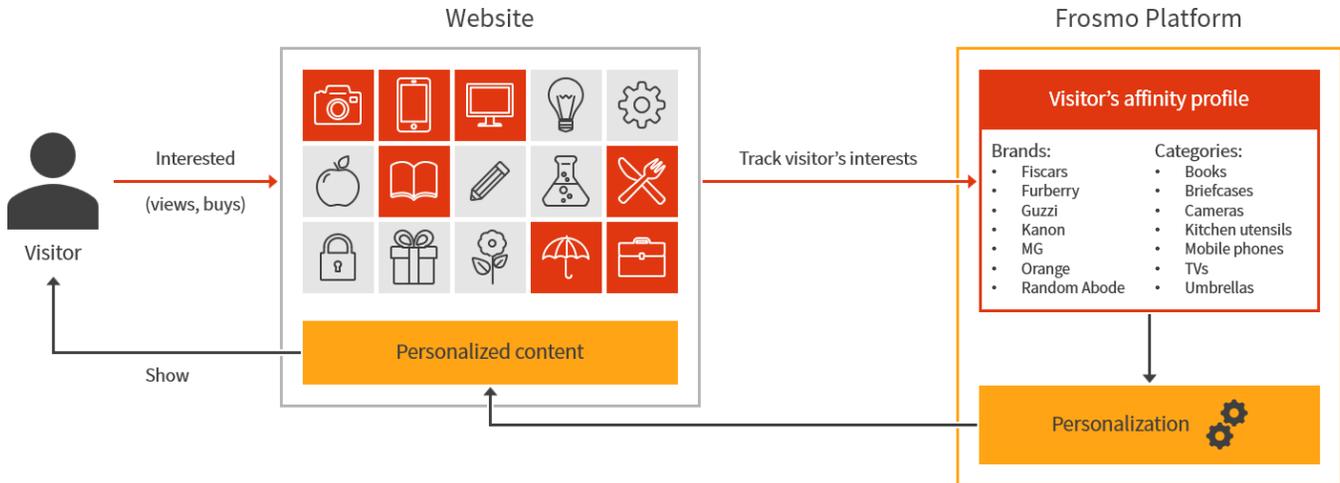


Figure: Affinities in one-to-one personalization (click to enlarge)

Affinity is measured along **item attributes**. A given affinity is always for a single attribute value, such as a specific category like "Power Tools/Drills" or "Roulette Games", rather than for a specific item. For example, a visitor can have an affinity for Wakita (brand) and an affinity for power drills (category), but not an affinity for a specific Wakita power drill (item). However, if you were to recommend power tools to the visitor, Wakita power drills would be a good match for the visitor's affinities. A visitor can have multiple affinities per item attribute. For example, the same visitor could have category affinities for "Hand Tools", "Power Tools/Drills", and "Power Tools/Saws". Each affinity has its own **affinity score**, which indicates the visitor's current level of affinity for that attribute value.

The Frosmo Platform automatically tracks affinities for each individual visitor in near real time. The totality of affinities tracked for a visitor make up the visitor's **affinity profile**. Whenever a visitor interacts with an item, for example by viewing a product page or launching a game, the platform updates the visitor's affinity profile accordingly. The more interactions, the more accurate and refined the profile. Moreover, recent interactions carry more weight than older ones. The profile therefore not only adjusts to the visitor's changing preferences over time, but also reflects what currently resonates with the visitor the most.

For more information about affinities, see:

- [Platform support for affinity](#)
- [Affinity scores](#)
- [Affinity profiles](#)
- [Affinity tracking](#)
- [Affinity-based recommendations](#)
- [Affinity example](#)
- [Where to go next](#)

Platform support for affinity

Learn about how the Frosmo Platform supports the tracking and use of affinity:

Supported attributes

The platform can track affinity for any item attribute whose type is string or array of strings.

The exact set of attributes tracked for affinity on a site is always separately planned and agreed with you. The platform can track up to 10 attributes for affinity per site.

For example, for a retail site, the platform might track brand, category, color, and tag for affinity. For an iGaming site, by comparison, the platform might track category, features, themes, vendor, and volatility.



The attributes tracked for affinity must be included in [product tracking](#). If the platform does not collect the attribute data from your site, there's nothing to measure affinity against.

Supported interactions

The platform tracks affinity based on the following types of visitor action:

- [Product view](#)
- [Transaction](#)

To track product views and transactions on a site, [product tracking](#) and [transaction tracking](#) must be correctly set up for the site.

Supported features

You can use affinity in the following features:

- [Recommendation strategy](#)

For more information about how affinity works in recommendations, see [Affinity-based recommendations](#).

You can also use [affinity groups](#) to target modifications to visitors.

Affinity scores

An **affinity score** is a numerical measure of how interested a visitor is in the items represented by an affinity. Each affinity has a single affinity score. The score is always a positive integer, such as 17 or 328. The higher the score, the greater the interest.

The Frosmo Platform considers the following factors when calculating the score for an affinity:

- **Interaction count:** The number of times the visitor has performed an action involving an item that matches the affinity. For example, the number of times the visitor has viewed items that belong to a specific brand. The platform counts all actions in the past 60 days that belong to the [supported types](#).



For transactions, the platform counts the number of separate transactions that an item has been a part of. How many copies or pieces of the item was purchased in a given transaction does not matter. For example, if a visitor purchases 1 piece of item A in one transaction, and then later 3 pieces of item A in another transaction, the platform counts this as 2 transactions for item A.

- **Interaction type:** Different types of actions have different weights when calculating the affinity score. For example, a transaction has more weight than a view, as transactions indicate greater engagement than views.
- **Recency:** Recent interactions have more weight than older interactions. The exact weight for a given day depends on how far back in the past that day is from the current day. The further back, the smaller the weight.

In calculating the score, the platform first calculates separate daily scores for the affinity based on the daily interactions, then applies recency weights to those scores, and finally combines the adjusted scores into the final affinity score.

Affinity profiles

An **affinity profile** is the full record of an individual visitor's current affinities on a site. The profile lists all affinities (attribute name and value) and corresponding [affinity scores](#) tracked for the visitor in the past 60 days.

For example, the affinity profile of a retail site visitor might look something like this:

| Attribute name | Attribute value | Affinity score |
|----------------|-------------------------|----------------|
| Brand | Cakita | 20 |
| Brand | Dawelt | 135 |
| Brand | Wakita | 328 |
| Category | Hand Tools/Pliers | 123 |
| Category | Hand Tools/Wrenches | 17 |
| Category | Power Tools/Accessories | 68 |

| | | |
|----------|--------------------|-----|
| Category | Power Tools/Drills | 170 |
| Category | Power Tools/Saws | 15 |

In the above example, the visitor has three brand affinities and five category affinities, with "Wakita" as the top brand affinity and "Power Tools/Drills" as the top category affinity.

The Frosmo Platform automatically creates and maintains a dedicated affinity profile for each individual visitor. The platform stores the profiles in the Frosmo back end. For more information about how this process works, see [Affinity tracking](#).

The platform saves the profiles as [JSON](#) objects. When returned by the platform, the above affinity profile would look like this:

```
{
  "brand": {
    "Cakita": 20,
    "Dawelt": 135,
    "Wakita": 328
  },
  "type": {
    "Hand Tools/Pliers": 123,
    "Hand Tools/Wrenches": 17,
    "Power Tools/Accessories": 68,
    "Power Tools/Drills": 170,
    "Power Tools/Saws": 15
  }
}
```



In the Frosmo back end, category is stored as the `type` attribute. For more information about item attributes, see the [product tracking developer guide](#).

You can view your own affinity profile in the **Visitor** section of [Frosmo Preview](#).

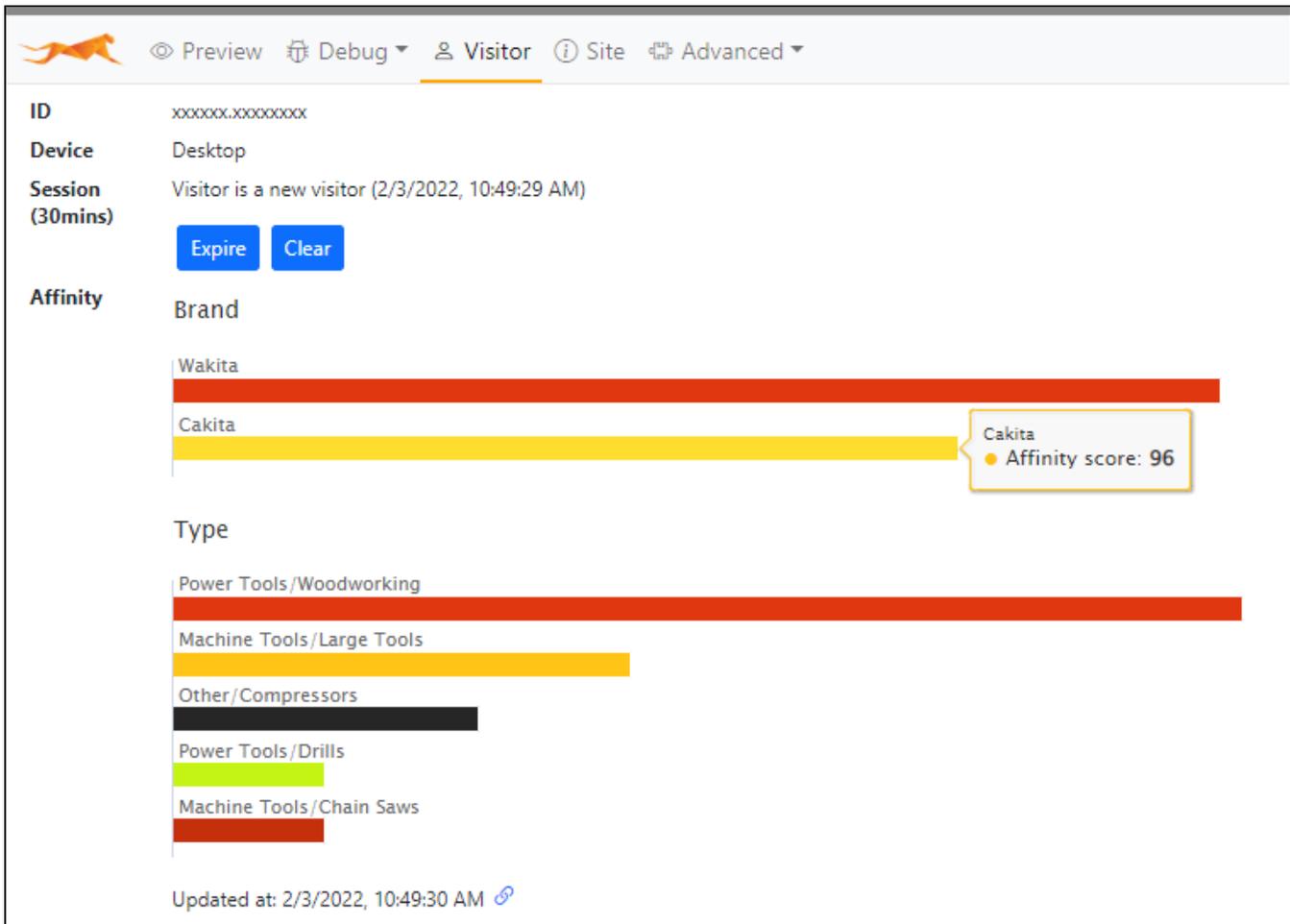


Figure: A visitor's affinity profile in Frosmo Preview

You can retrieve any visitor's affinity profile from the Frosmo back end with the [Visitor API](#).

Affinity tracking

The Frosmo Platform automatically tracks affinities for every visitor on a site. You do not need to separately set up affinity tracking for a site. However, for the affinity tracking to work, the site must have [product tracking](#) and [transaction tracking](#) correctly set up.

Affinity tracking works as follows:

1. On a site, a visitor does one of the following:
 - Starts a new [session](#), triggering a visit event. The visitor is either a new visitor or a returning visitor whose previous session has expired.
 - Performs an [action that indicates affinity](#). The action triggers a corresponding data tracking event. For example: The visitor navigates to a product page, triggering a product view event.
2. [Frosmo Core](#) sends the event data to the Frosmo back end.
3. The platform updates the visitor's [affinity profile](#):
 - The platform recalculates the scores of all affinities in the profile.
 - The platform adds new affinities to the profile, if any.
 - The platform removes obsolete affinities from the profile, if any. An obsolete affinity is one that is based on events older than 60 days.

If the visitor does not yet have an affinity profile, the platform creates a new profile for them.

4. Once a day, the platform goes through the event data for all visitors and removes events older than 60 days. An affinity that is based on these events is automatically removed from a visitor's profile the next time the platform updates the profile.

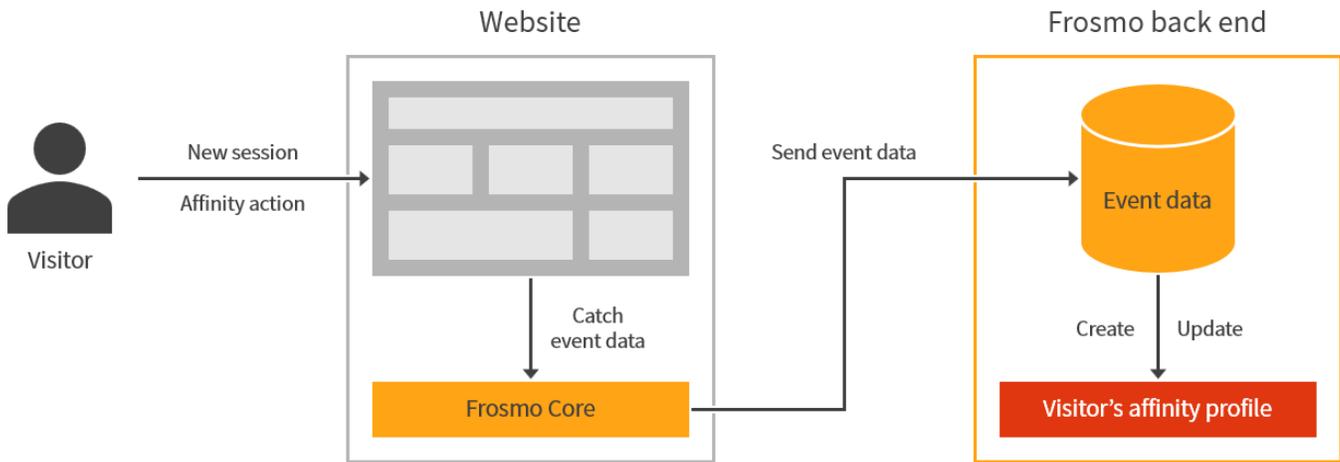


Figure: Affinity tracking in the Frosmo Platform (click to enlarge)

Affinity-based recommendations

In a [recommendation](#), affinity reorders (reranks) the recommendation results for each visitor based on their affinity profile. Each visitor thus sees a different version of the same recommendation, one tailored to their individual interests and preferences.

 Affinities only work in recommendations based on [recommendation strategies](#).

You can set the level of affinity applied per recommendation strategy. The higher the level, the greater the weight given to affinities, and the more one-to-one-personalized the set of recommended items returned by the strategy. For more information about setting the level, see [Creating and editing a recommendation strategy](#).

The following figures show an example of the same affinity-based recommendation as displayed to two different visitors on a retail site. The first visitor has an affinity for drills, while the second visitor has an affinity for fruits and vegetables.

Most popular for you

| | | | | |
|---|--|---|---|--|
|  |  |  | <div style="background-color: red; color: white; padding: 2px; font-weight: bold; font-size: 0.8em;">SALE</div>  | <div style="background-color: red; color: white; padding: 2px; font-weight: bold; font-size: 0.8em;">SALE</div>  |
| Brandix Drill Screwdriver ALX7054 200 Watts €850.00 <div style="background-color: black; color: white; padding: 5px; font-weight: bold;">Add to cart</div> In stock, 9 left | Brandix Drill Series 3 KSR4590PQS 1500 Watts €949.00 <div style="background-color: black; color: white; padding: 5px; font-weight: bold;">Add to cart</div> In stock, 4 left | Brandix Screwdriver SCREW1500ACC €1499.00 <div style="background-color: black; color: white; padding: 5px; font-weight: bold;">Add to cart</div> In stock, 5 left | Brandix Angle Grinder KZX3890PQW €525.69 €649.00 <div style="background-color: black; color: white; padding: 5px; font-weight: bold;">Add to cart</div> In stock, 5 left | Brandix Electric Planer KL370090G 300 Watts €509.32 €749.00 <div style="background-color: black; color: white; padding: 5px; font-weight: bold;">Add to cart</div> In stock, 7 left |

Figure: Recommendation personalized for a visitor with an affinity for drills (click to enlarge)

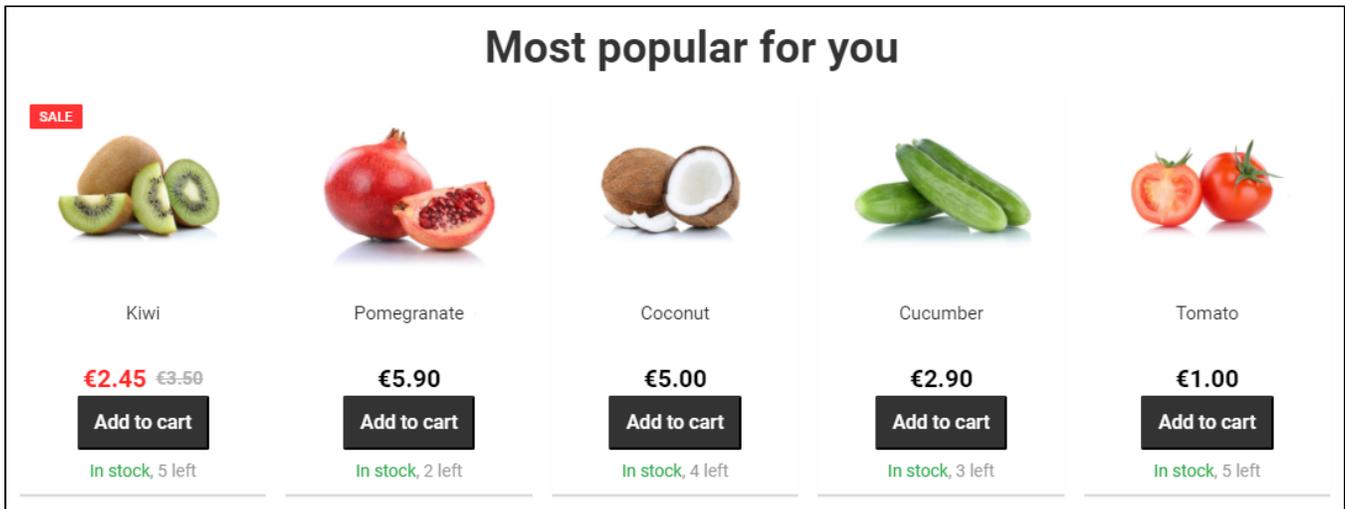


Figure: Recommendation personalized for a visitor with an affinity for fruits and vegetables (click to enlarge)

Generating an affinity-based recommendation

The process for generating an affinity-based recommendation flows as follows:

1. The platform generates the [recommendation data](#) separately for each [algorithm](#) in the recommendation strategy.

The platform returns a maximum of 120 items per algorithm. The items are in descending order of rank, with the most recommended item (as defined by the algorithm) ranked highest.

2. The platform applies [filters](#), if defined, to the recommendation data.

The filters remove items from the data, assuming any items match the filter rules. The filtered data forms the baseline set of items from which the platform picks the final items returned by the strategy.

3. The platform applies affinity to the recommendation data:
 - a. The platform fetches the visitor's [affinity profile](#).
 - b. The platform separately reranks each algorithm's recommendation data based on the visitor's affinities. How extensive the reranking is depends on the level of affinity set in the strategy.

- b. The platform separately reranks each algorithm's recommendation data based on the visitor's affinities. How extensive the reranking is depends on the level of affinity set in the strategy.

4. The platform picks the top items for each algorithm. The number of items picked depends on the algorithm settings in the strategy.

5. The platform combines the top items and any [fixed items](#) into the final set of recommended items returned by the strategy.

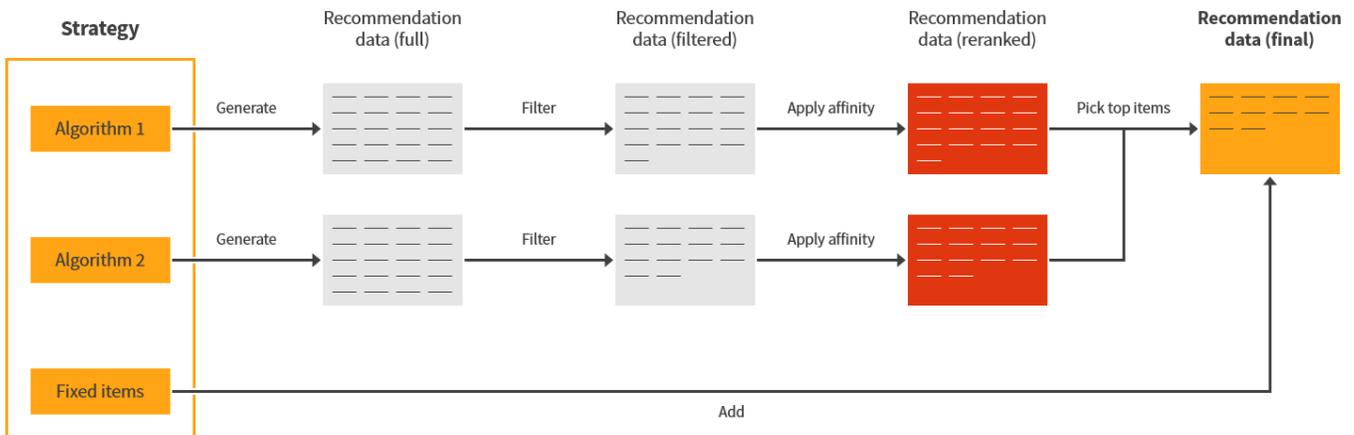


Figure: Generating an affinity-based recommendation (click to enlarge)

Calculating affinity for a recommendation

Applying affinity to recommendation data ([step 3 above](#)) means taking the original ranking score of each item (as defined by an algorithm) and combining that score with an affinity score calculated for the item. The item affinity score is based on a visitor's affinity profile. The level of affinity set in the strategy determines the relative weights of the two scores. The combined score defines the item's final ranking for the visitor.

If two items get the exact same final score, the platform determines their order randomly.

Affinity example

This example walks you through on a high level how the Frosmo Platform tracks a single visitor's affinity over time and applies the resulting affinity profile to a recommendation displayed to the visitor.

- [Supported attributes](#)
- [Supported interactions](#)
- [Supported features](#)
- [Generating an affinity-based recommendation](#)
- [Calculating affinity for a recommendation](#)
- [Assumptions for simplicity](#)
- [Tracking the visitor's affinity over time](#)
- [Displaying an affinity-based recommendation to the visitor](#)

Assumptions for simplicity

To keep things simple for example purposes, let's assume the following:

- The platform tracks affinity for brands and categories.
- The platform tracks affinity based on product views and transactions. By default, a single view is worth 1 point of affinity, while a single transaction is worth 3 points of affinity. One transaction is thus worth three views.
- The platform distinguishes between three time frames of recency: today, last week, and last two months (60 days). Interactions within these time frames are weighted at 10x, 5x, and 1x, respectively. For example, a single transaction on the current day is worth $10 \text{ (recency weight)} \times 3 \text{ (interaction type)} \times 1 \text{ (interaction count)} = 30$ points of affinity, while a single transaction from two days ago is worth $5 \text{ (recency weight)} \times 3 \text{ (interaction type)} \times 1 \text{ (interaction count)} = 15$ points of affinity.
- The platform ignores interactions older than 60 days.



The above assumptions – and this example at large – represent a simplified version of how the platform actually tracks and calculates affinity. Even so, this example gives you a roughly representative picture of how the platform works.

Tracking the visitor's affinity over time

Let's call our visitor Alex. Alex is interested in power drills and shoes. Alex visits a retail site on three separate days (August 1st, August 5th, and October 2nd), logging a single session each day.

Day 1

On August 1st, Alex only views items. Alex's first view is of a Wakita power drill, as shown in the following table.

Table: Alex's first interaction on day 1

| Item | Brand | Category | Views | Transactions |
|----------------------------------|--------|--------------------|-------|--------------|
| Wakita Drill Screwdriver ALX7054 | Wakita | Power Tools/Drills | 1 | 0 |

Based on the interaction, the platform creates the following affinity profile for Alex.

Table: Alex's affinity profile after first interaction on day 1

| Attribute name | Attribute value | Affinity score |
|----------------|--------------------|----------------|
| Brand | Wakita | 10 |
| Category | Power Tools/Drills | 10 |

Since a single view is worth 1 point of affinity, and since interactions for the current day are weighted at 10x, this gives you an affinity score of $10 \times 1 \times 1 = 10$ for the power drill's brand and category.

Alex then proceeds to view a few more items during their session. The following table lists (in alphabetical order) all the items Alex has interacted with by the end of the session as well as the interaction counts per item.

Table: Alex's interactions on day 1

| Item | Brand | Category | Views | Transactions |
|--------------------------------------|------------|-----------------------|-------|--------------|
| Kosch Drill Screwdriver SCREW1500ACC | Kosch | Power Tools/Drills | 1 | 0 |
| Lumberland Casual Engineer Boots | Lumberland | Women's Fashion/Shoes | 2 | 0 |
| Perly Lace-up Leather Boots | Perly | Women's Fashion/Shoes | 3 | 0 |

| | | | | |
|----------------------------------|--------|--------------------|---|---|
| Wakita Drill Screwdriver ALX7054 | Wakita | Power Tools/Drills | 2 | 0 |
| Wakita Drill Series 3 KSR4590PQS | Wakita | Power Tools/Drills | 1 | 0 |

The following table shows Alex's affinity profile at the end of day 1, updated to include all the interactions from their first session on the site.

Table: Alex's affinity profile at the end of day 1

| Attribute name | Attribute value | Affinity score |
|----------------|-----------------------|----------------|
| Brand | Kosch | 10 |
| Brand | Lumberland | 20 |
| Brand | Perly | 30 |
| Brand | Wakita | 30 |
| Category | Power Tools/Drills | 40 |
| Category | Women's Fashion/Shoes | 50 |

For example, Alex viewed items belonging to the "Power Tools/Drills" category a total of four times, which gives you an affinity score of $10 \times 1 \times 4 = 40$ for that category.

Day 2

On August 5th, when Alex returns to the site, but before they interact with any items, the platform automatically updates their affinity profile. At this point, Alex's affinity profile is based entirely on interactions from day 1, August 1st.

The affinity scores from day 1 are recalculated with a recency weight of 5 (last week). For example, the four views for "Power Tools/Drills" items now amount to an affinity score of $5 \times 1 \times 4 = 20$ for the category.

Table: Alex's affinity profile at the beginning of day 2

| Attribute name | Attribute value | Affinity score |
|----------------|-----------------------|----------------|
| Brand | Kosch | 5 |
| Brand | Lumberland | 10 |
| Brand | Perly | 15 |
| Brand | Wakita | 15 |
| Category | Power Tools/Drills | 20 |
| Category | Women's Fashion/Shoes | 25 |

During day 2, Alex buys two items, also viewing them once in the process.

Table: Alex's interactions on day 2

| Item | Brand | Category | Views | Transactions |
|----------------------------------|--------|-----------------------|-------|--------------|
| Perly Lace-up Leather Boots | Perly | Women's Fashion/Shoes | 1 | 1 |
| Wakita Drill Screwdriver ALX7054 | Wakita | Power Tools/Drills | 1 | 1 |

The following table shows Alex's affinity profile at the end of day 2, updated to include all the interactions from their second session on the site.

Table: Alex's affinity profile at the end of day 2

| Attribute name | Attribute value | Affinity score |
|----------------|--------------------|----------------|
| Brand | Kosch | 5 |
| Brand | Lumberland | 10 |
| Brand | Perly | 55 |
| Brand | Wakita | 55 |
| Category | Power Tools/Drills | 60 |

| | | |
|----------|-----------------------|----|
| Category | Women's Fashion/Shoes | 65 |
|----------|-----------------------|----|

Day 2 does not introduce any new affinities to the profile, but it does significantly boost revisited affinities. The affinity scores for day 2 are calculated with a recency weight of 10 (today) and the resulting scores are added to the corresponding recalculated affinity scores from day 1. For example, the "Power Tools/Drills" category got one view and one transaction on day 2, which gives it a daily affinity score of $(10 \times 1 \times 1) + (10 \times 3 \times 1) = 40$. Add that to the recalculated daily score from day 1, and you get a total current affinity score of $20 + 40 = 60$ for the category.

Day 3

On October 2nd, when Alex comes back to the site after a long break, but before they interact with any items, the platform again automatically updates their affinity profile. Since it's now more than 60 days from day 1, August 1st, all interactions from that day are omitted. Alex's tracked affinities at this point are based entirely on interactions from day 2, August 5th.

On day 2, each affinity got one view and one transaction, which now, almost 60 days later, translate into an affinity score of $(1 \times 1 \times 1) + (1 \times 3 \times 1) = 4$. The recency weight is 1 (last two months).

Table: Alex's affinity profile at the beginning of day 3

| Attribute name | Attribute value | Affinity score |
|----------------|-----------------------|----------------|
| Brand | Perly | 4 |
| Brand | Wakita | 4 |
| Category | Power Tools/Drills | 4 |
| Category | Women's Fashion/Shoes | 4 |

During day 3, Alex views three items and buys two of them.

Table: Alex's interactions on day 3

| Item | Brand | Category | Views | Transactions |
|----------------------------------|------------|-------------------------|-------|--------------|
| Lumberland Casual Engineer Boots | Lumberland | Women's Fashion/Shoes | 1 | 1 |
| Wakita Battery Pack 20V 5.0Ah | Wakita | Power Tools/Accessories | 2 | 0 |
| Wakita Metal Drill Bit Set 29 | Wakita | Power Tools/Accessories | 3 | 1 |

The following table shows Alex's affinity profile at the end of day 3, updated to include all the interactions from their third session on the site.

Table: Alex's affinity profile at the end of day 3

| Attribute name | Attribute value | Affinity score |
|----------------|-------------------------|----------------|
| Brand | Lumberland | 40 |
| Brand | Perly | 4 |
| Brand | Wakita | 84 |
| Category | Power Tools/Accessories | 80 |
| Category | Power Tools/Drills | 4 |
| Category | Women's Fashion/Shoes | 44 |

Since Alex does not view or buy any "Perly" or "Power Tools/Drills" items, the corresponding affinities remain at their recalculated scores from the start of the day (4). The remaining affinities, however, two carried over from day 2 ("Wakita", "Women's Fashion/Shoes") and two introduced on day 3 ("Lumberland", "Power Tools/Accessories"), all get a boost from Alex's interactions on day 3. For example:

- The "Wakita" brand gets a total of five views and one transaction on day 3, which gives you an affinity score of $(10 \times 1 \times 5) + (10 \times 3 \times 1) = 80$ for that day. Combined with the recalculated score from day 2, the day 3 score pushes Alex's total affinity score for the brand to $4 + 80 = 84$.
- The "Power Tools/Accessories" category similarly gets a total of five views and one transaction on day 3, resulting in the same daily affinity score of 80. However, since Alex's day 3 interactions are the only interactions tracked for the category, the day 3 score is also Alex's total affinity score for the category.

Displaying an affinity-based recommendation to the visitor

For example purposes, let's assume that our retail site displays a simple "Most popular for you" recommendation to all visitors on multiple pages. The recommendation uses a [strategy](#) that returns the 5 most viewed items on the site in the past 7 days with affinity applied. The strategy does not use fixed items or filters.

Let's further assume that the algorithm used by the strategy returns the following ten items before the strategy applies affinity and picks the top five items for display.

Table: Items returned by the recommendation algorithm

| Slot | Item | Brand | Category |
|------|--------------------------------------|------------|-----------------------|
| 1 | Plastic bucket | Frosmo | Home/Accessories |
| 2 | Stiffa Sneakers 2021 | Stiffa | Kids' Fashion/Shoes |
| 3 | Kosch Drill Screwdriver SCREW1500ACC | Kosch | Power Tools/Drills |
| 4 | Cakita Circular Saw DPS3000SY | Cakita | Power Tools/Saws |
| 5 | Lumberland Comfort Logger Boots | Lumberland | Men's Fashion/Shoes |
| 6 | Dawelt Drill DXD699B | Dawelt | Power Tools/Drills |
| 7 | Wakita Pliers | Wakita | Hand Tools/Pliers |
| 8 | Perly Bum Bag | Perly | Men's Fashion/Bags |
| 9 | Stiffa Sneakers 2020 | Stiffa | Women's Fashion/Shoes |
| 10 | Wakita Drill Screwdriver ALX7054 | Wakita | Power Tools/Drills |

Which five items Alex actually sees displayed in the recommendation and in what order depends on:

- Alex's affinity profile at the moment the recommendation loads on the page
- The level of affinity the strategy is set to apply

Let's take Alex's affinity profile from the end of day 2 as our example.

Table: Alex's affinity profile at the end of day 2

| Attribute name | Attribute value | Affinity score |
|----------------|-----------------------|----------------|
| Brand | Kosch | 5 |
| Brand | Lumberland | 10 |
| Brand | Perly | 55 |
| Brand | Wakita | 55 |
| Category | Power Tools/Drills | 60 |
| Category | Women's Fashion/Shoes | 65 |

When applying a visitor's affinities to a recommendation, the platform first calculates an affinity score for each item based on the visitor's affinity profile, and then reranks the items based on the level of affinity defined in the strategy. Here, a single item's affinity score is a function of the visitor's brand and category affinity scores that match the item's brand and category.

For example purposes, let's look at Alex's final ranking scores at two levels of affinity: 50% and 100%. The former gives equal weight to the original ranking scores and the item affinity scores, while the latter ignores the original ranking scores and reranks the items purely based on the item affinity scores.

The following table shows a side-by-side comparison of what the reranked sets of recommended items might look like for Alex at 50% affinity and 100% affinity.

Table: Items returned by the recommendation algorithm, reranked for Alex at 50% and 100% affinity

| Slot | 50% affinity | 100% affinity |
|------|--------------------------------------|--------------------------------------|
| 1 | Kosch Drill Screwdriver SCREW1500ACC | Wakita Drill Screwdriver ALX7054 |
| 2 | Wakita Drill Screwdriver ALX7054 | Kosch Drill Screwdriver SCREW1500ACC |
| 3 | Plastic bucket | Stiffa Sneakers 2020 |
| 4 | Dawelt Drill DXD699B | Dawelt Drill DXD699B |
| 5 | Stiffa Sneakers 2021 | Wakita Pliers |
| 6 | Stiffa Sneakers 2020 | Perly Bum Bag |
| 7 | Wakita Pliers | Lumberland Comfort Logger Boots |

| | | |
|----|---------------------------------|-------------------------------|
| 8 | Perly Bum Bag | Plastic bucket |
| 9 | Cakita Circular Saw DPS3000SY | Stiffa Sneakers 2021 |
| 10 | Lumberland Comfort Logger Boots | Cakita Circular Saw DPS3000SY |

Finally, the following table shows a side-by-side comparison of the top 5 items Alex sees in the recommendation in the following cases: no affinity applied, 50% affinity applied, and 100% affinity applied.

Table: Final recommended items for Alex at no affinity, 50% affinity, and 100% affinity

| Slot | No affinity | 50% affinity | 100% affinity |
|------|--------------------------------------|--------------------------------------|--------------------------------------|
| 1 | Plastic bucket | Kosch Drill Screwdriver SCREW1500ACC | Wakita Drill Screwdriver ALX7054 |
| 2 | Stiffa Sneakers 2021 | Wakita Drill Screwdriver ALX7054 | Kosch Drill Screwdriver SCREW1500ACC |
| 3 | Kosch Drill Screwdriver SCREW1500ACC | Plastic bucket | Stiffa Sneakers 2020 |
| 4 | Cakita Circular Saw DPS3000SY | Dawelt Drill DXD699B | Dawelt Drill DXD699B |
| 5 | Lumberland Comfort Logger Boots | Stiffa Sneakers 2021 | Wakita Pliers |

And that's our affinity example. Phew!

Where to go next

- Learn how to [create and edit recommendation strategies](#).
- Learn from a [practical example of creating a recommendation based on affinity](#).
- Learn about [affinity groups](#).
- If you're a developer, learn how to [fetch recommendation data generated from recommendation strategies](#).
- If you're a developer, learn how to [fetch a visitor's affinity profile](#).