

Days on Market

Understand computation logic for DOM.

Revision history

- 30-Jan-2023:
 - Corrected error in listing's merge step in all 3 examples
 - Removed 3 slides of dom algo each explaining dom_active_is, dom_180_is and dom_is. Added one generic algo that describes all 3 types.
 - Changed the typo in field name dom_active_is.

There are three fields related with Days On Market :

1. **dom_active_is** : DOM of a car considering the last 30 days period
2. **dom_180_is** : DOM of a car considering only the last 180 days period
3. **dom_is** : DOM of a car for its lifetime

DOM computation logic

1. Consider all listings of a car that were online in the last N days,
where,
N = 30 for dom_active_is
N = 180 for dom_180_is
N = infinite for dom_is
2. Sort the listings on status_date descending order, i.e. latest updated listings are on top.
3. Then merge all overlapping intervals of (scraped_at_date, status_date). Repeat the merging operation until no overlapping intervals based on dates are obtained. At each merge operation, the miles values are updated with the average of the miles of the listings used for merging.
4. We assume that the vehicle's miles should change at least 500 miles before appearing back on the market. Therefore the gaps in terms of number of days between the intervals are also merged provided their miles values differ not more than 500 miles. Repeat the merging operation until no overlapping intervals based on miles are obtained. At each merge operation, the miles values are updated with the average of the miles of the listings used for merging.
5. The summation of the number of days in each merged interval gives the respective DOM value.

Let us understand calculation of these 3 fields with the help of an example.

Consider the following table as example listings data from historical database.

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	dealer1	(d-12)	(d-1)	10000
Vin1	dealer2	(d-05)	(d)	10010
Vin1	dealer3	(d-40)	(d-10)	9500
Vin1	dealer4	(d-100)	(d-90)	8000
Vin1	dealer5	(d-150)	(d-120)	7950
Vin1	dealer5	(d-190)	(d-170)	7000
Vin1	dealer1	(d-500)	(d-400)	1000

Here (d) is the current day and (d-n) indicates n days prior to current day.

1. Computing *dom_is_active*

1. Consider all listings of a car that were online in the last 30 days. With this rule the following listings get selected.

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	dealer1	(d-12)	(d-1)	10000
Vin1	dealer2	(d-05)	(d)	10010
Vin1	dealer3	(d-40)	(d-10)	9500

2. Sort the listings on status_date descending order, i.e. latest updated listings are on top.

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	dealer2	(d-05)	(d)	10010
Vin1	dealer1	(d-12)	(d-1)	10000
Vin1	dealer3	(d-40)	(d-10)	9500

3. Then merge all overlapping intervals of (scraped_at_date, status_date) repeatedly. Notice the dealer is replaced with xxxxx, this is done as we are interested in the "**Market**" perspective, so which dealer the listing was active upon is ignored.

Listings after the first step of merging

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	xxxxx	(d-12)	(d)	10005
Vin1	xxxxx	(d-40)	(d-10)	9500

Notice the miles value is the average of two intervals

Listings after the second step of merging

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	xxxxx	(d-40)	(d)	9752.5

4. Get interval duration in terms number of days for each interval

VIN	Dealer	scraped_at_date	status_date	miles	Calculated number of days
Vin1	xxxxx	(d-40)	(d)	9752.5	40

5. Total dom_active_is is a summation of above values which is **40** days.

2. Computing ***dom_180_is***

1. Consider all listings of a car that were online in the last 180 days. With this rule the following listings get selected.

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	dealer1	(d-12)	(d-1)	10000
Vin1	dealer2	(d-05)	(d)	10010
Vin1	dealer3	(d-40)	(d-10)	9500
Vin1	dealer4	(d-100)	(d-90)	8000
Vin1	dealer5	(d-150)	(d-120)	7950
Vin1	dealer5	(d-190)	(d-170)	7000

2. Sort the listings on status_date descending order, i.e. latest updated listings are on top.

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	dealer2	(d-05)	(d)	10010
Vin1	dealer1	(d-12)	(d-1)	10000
Vin1	dealer3	(d-40)	(d-10)	9500
Vin1	dealer4	(d-100)	(d-90)	8000
Vin1	dealer5	(d-150)	(d-120)	7950
Vin1	dealer5	(d-190)	(d-170)	7000

3. Then merge all overlapping intervals of (scraped_at_date, status_date) repeatedly.

Notice the miles value is the average of all the intervals

After all repeated iterations based on dates the merged listings look like this

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	xxxxxx	(d-40)	(d)	9752.5
Vin1	xxxxxx	(d-100)	(d-90)	8000
Vin1	xxxxxx	(d-150)	(d-120)	7950
Vin1	xxxxxx	(d-190)	(d-170)	7000

4. Now, merging intervals with miles difference not more than 500.

Notice the miles value is the average of two intervals and intervals extended

VIN	Dealer	scraped_at_date	status_date	miles
Vin1	xxxxxx	(d-12)	(d)	10005
Vin1	xxxxxx	(d-40)	(d-10)	9500
Vin1	xxxxxx	(d-150)	(d-90)	7975
Vin1	xxxxxx	(d-190)	(d-170)	7000

this interval is **NOT** merged as the miles difference is more than 500 miles with its adjacent listing

5. Get interval duration in terms number of days for each interval

VIN	Dealer	scraped_at_date	status_date	miles	Calculated number of days
Vin1	xxxxxx	(d-40)	(d)	9752.5	40
Vin1	xxxxxx	(d-150)	(d-90)	7975	40
Vin1	xxxxxx	(d-190)	(d-170)	7000	20

6. Total dom_180_is is a summation of above values which is **100** days.

3. Computing ***dom_is***

1. **dom_is** is similar to ***dom_180_is*** calculation. We need to consider the last listing and calculate interval duration in terms of number of days

VIN	Dealer	scraped_at_date	status_date	miles	Calculated number of days
Vin1	dealer1	(d-500)	(d-400)	1000	100

So **dom_is = 200 days**