Ranking Factors:

Going beyond just relevance by transforming 🪨 → 💎

~~Best~~ Business Match (5 min)

Signals in the rough (5 min)

Cut and polish signal gems (10 min)

Blend them beautifully (5 min)

Nate Day @natedayta
OpenSource Connections @o19s
Mices EU
2022-06-15

Best Match is out…Business Match is in

Best Match

☠️ Only for search nerds

☠️ Term Frequency

- Keyword stuffing

☠️ Length norms

- Favor shorter, less descriptive fields

Incentivizes bad behavior for content creators

Business Match

👼 Extends beyond just the search team

- Business objectives are plentiful and well defined

👼 Search as an organization level effort

- Merchandising, brand relationship, executive strategy intersect here

Improved search understanding & involvement across the entire org

YOUR DEPARTMENT GETS A KPI
EVERY DEPARTMENT GETS A KPI!
imgflip.com

“A diamond in the rough” 💎

Idiom

Something that is in poor condition but that is likely to become valuable with appropriate care or attention.

example: The house is a diamond in the rough, and with some hard work it will be really beautiful

Signals in the rough

Counts
Clicks
Purchases
Returns
Stock

Continuous
Price
Ratings

Qualitative
Curator picks
Brand tiers

IF YOU USE USE
LIFETIME RANKING SINGNALS
NSTRUCTO
YOU'RE GONNA HAVE A BAD TIME
imgflip.com

Nothing gold 👑 can stay

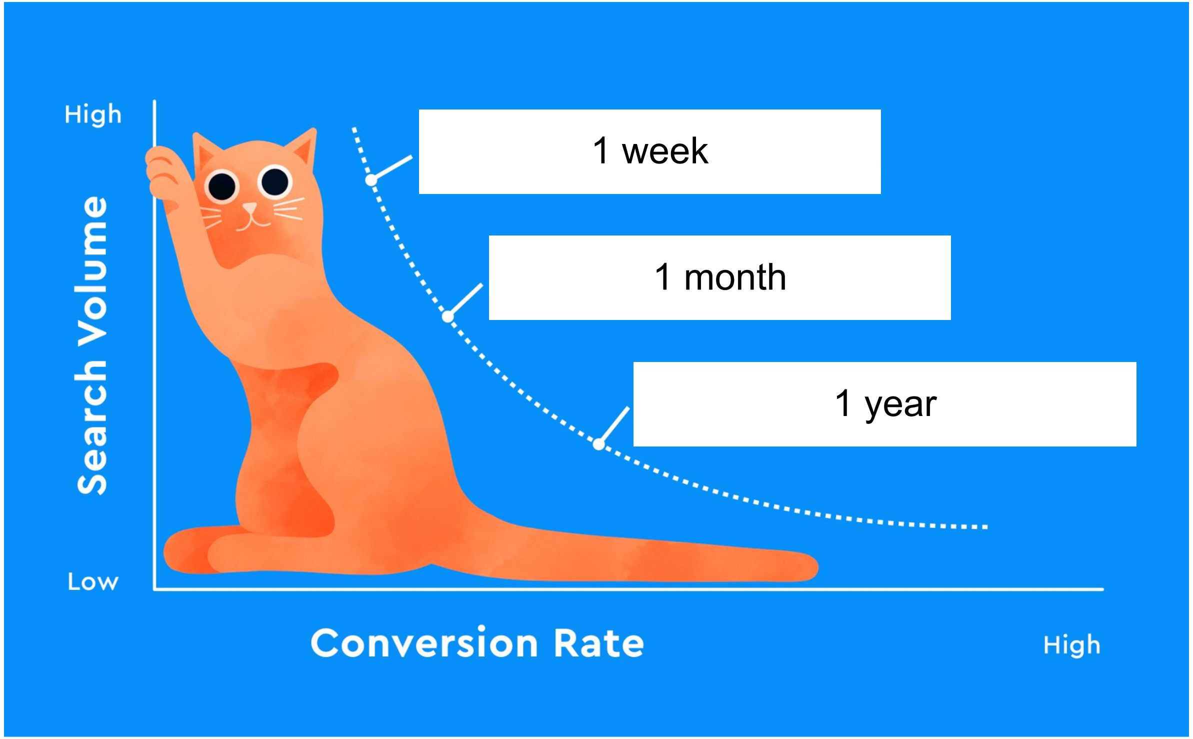
A poem by Robert Frost:

Nature’s first green is gold,
Her hardest hue to hold.
Her early leaf’s a flower,
But only so an hour.
Then leaf subsides to leaf.
So Eden sank to grief,
So dawn goes down to day.
Nothing gold can stay.

All ranking factors should be computed for a window in time

e.g. 1 week, 1 month, 1 quarter

Scale time windows by search volume



Uncut gems → ⚡ → Priceless pieces

Raw values are unbounded

- similar to BM25

Makes combining multiple factors difficult

Goal: scale values into a pre-set range

- [0,1] and [-1,1] make life easy

Useful scalers: max

Shifts values to the range [-∞,1]

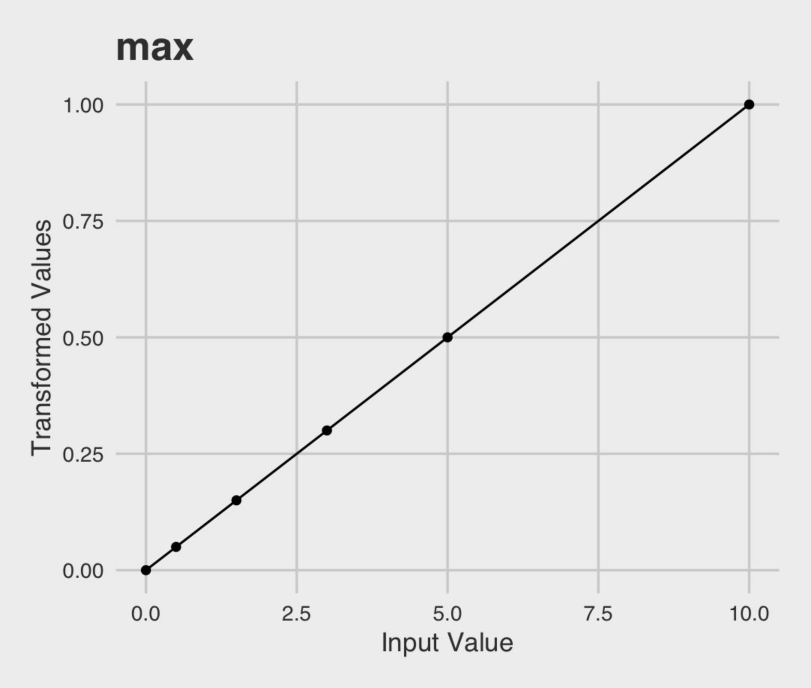
Relative distance are preserved

Great for scenarios were the minimum is unknown

- like BM25

```
max = function(x) {
  max_x = max(x)

  x / max_x
}
```



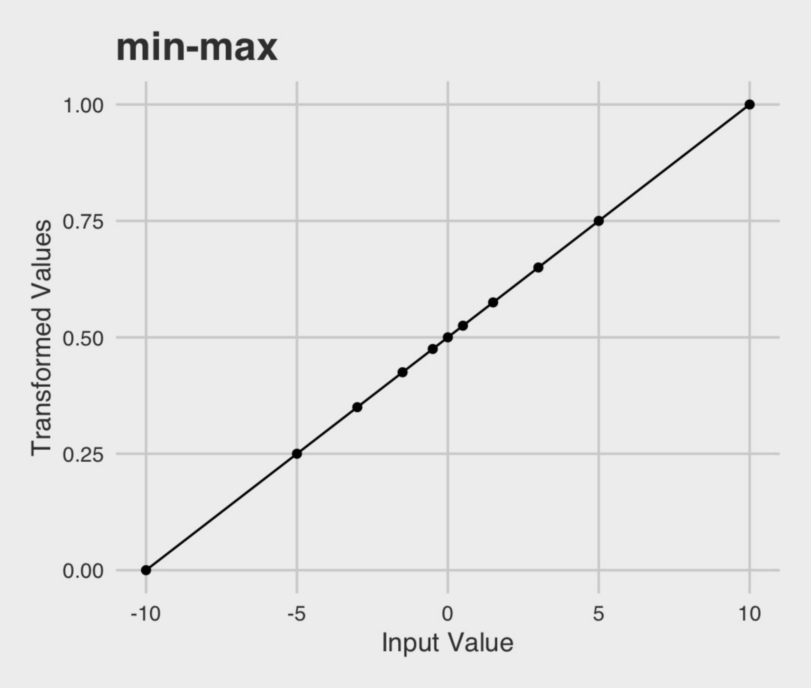
Useful normalizations: min-max

Shifts values to the range [0,1]

Relative distance are preserved

```
min_max = function(x) {
  min_x = min(x)
  max_x = max(x)

  (x - min_x) / (max_x -
min_x)
}
```



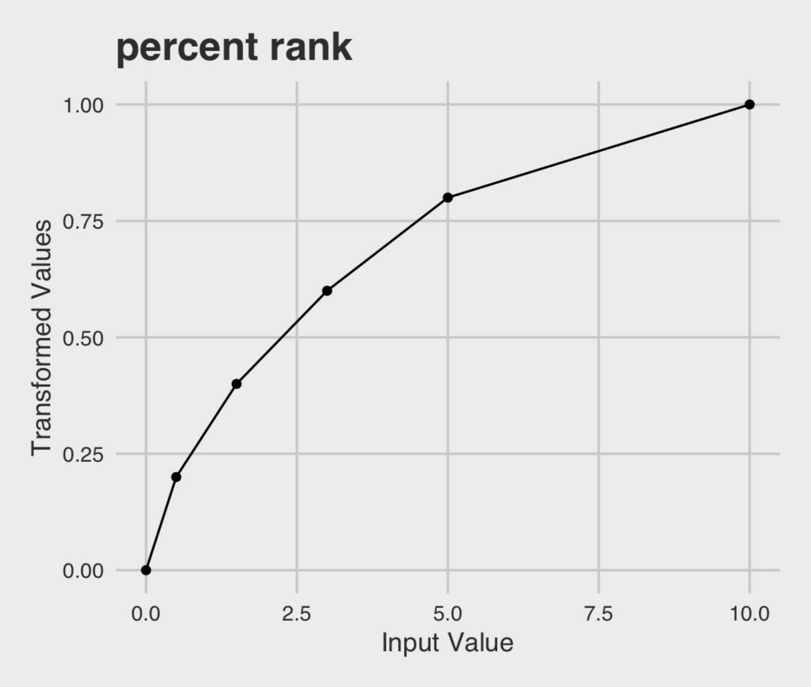
Useful normalizations: percent rank

Shifts values to the range [0,1]

Reflects rank order

Loses relative distance

```
perecent_rank = function(x)
{
  (min_rank(x) - 1) /
  (sum(x) - 1)
}
```



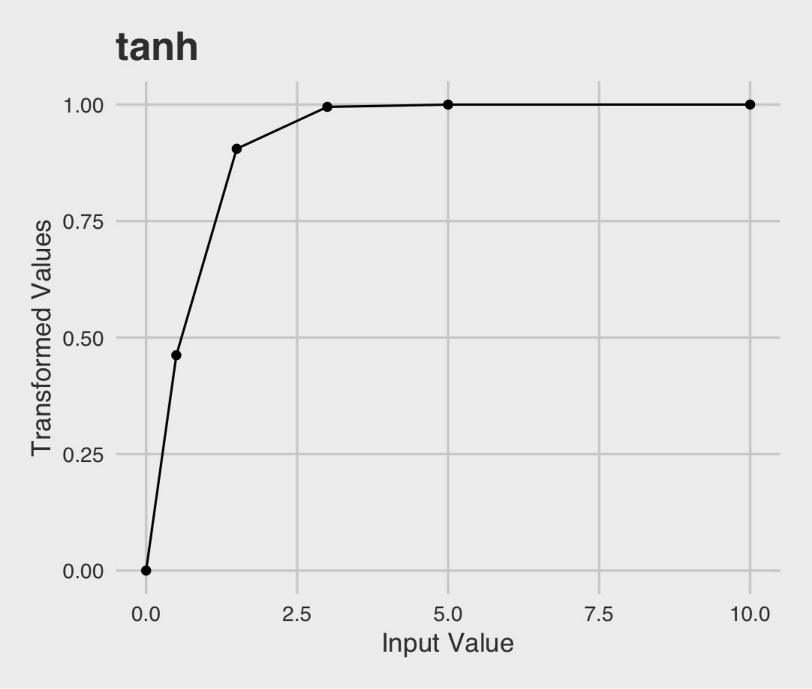
Useful normalizations: tanh

Shifts values to the range [-1, 1]

- positive values shift to [0, 1]

“Smushes” extreme values more

- Loses relative distance



@natedayta / @o19s

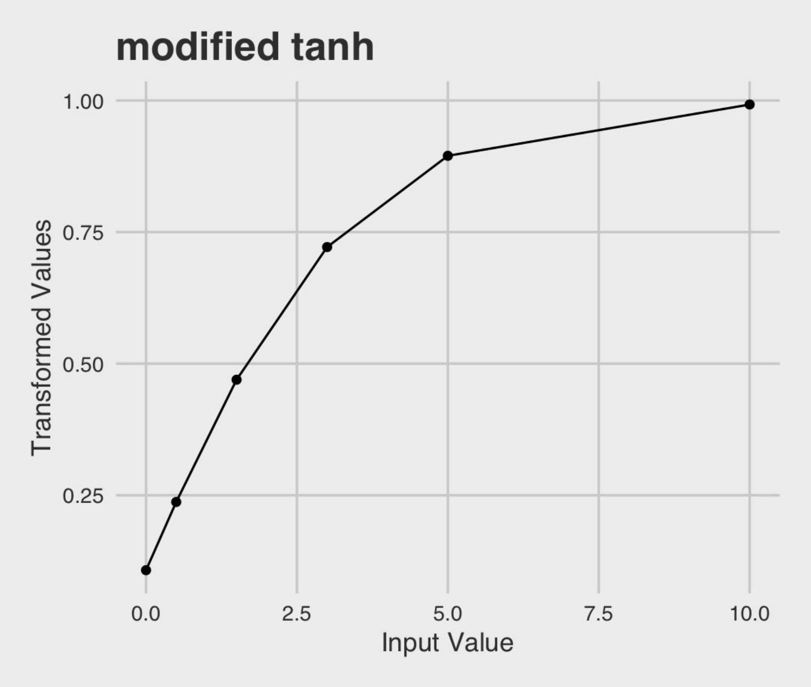
Useful normalizations: modified tanh

Control the amount of “smush”

Input values of zero are indeterminate

```
modified_tanh = function(x) {
  mu_x = mean(x)
  sd_x = sd(x)
  upper_bound = 1

  upper_bound *
tanh(1*((x-mu_x)/sd_x) + 1)
}
```



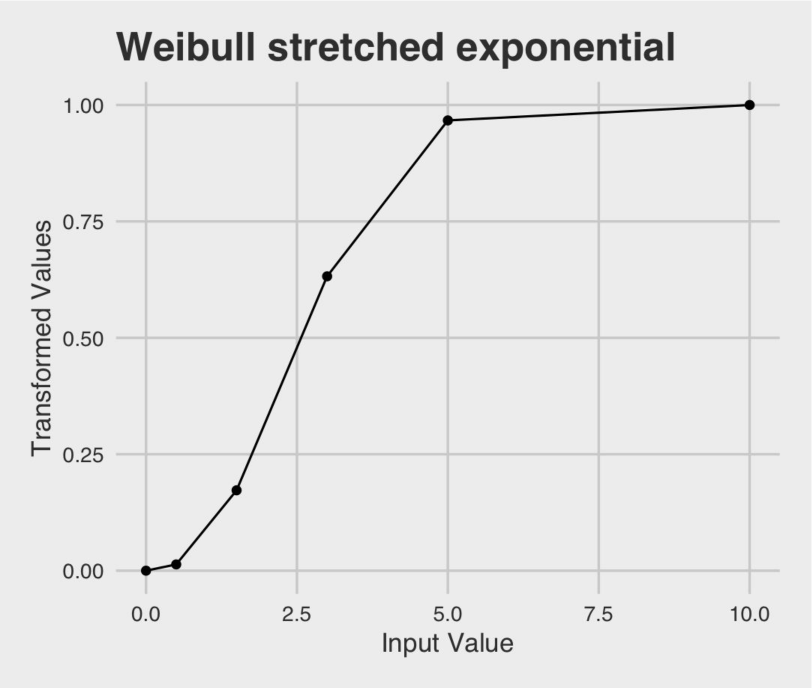
Useful normalizations: Weibull "stretched exponential"

Control the “smush” factor independently at either end

- varA & varB are the control parameters

```
stretch_weibull = function(x) {
  var_a = 3
  var_b = 2.4

  1 - exp(-(x/var_a)^var_b)
}
```

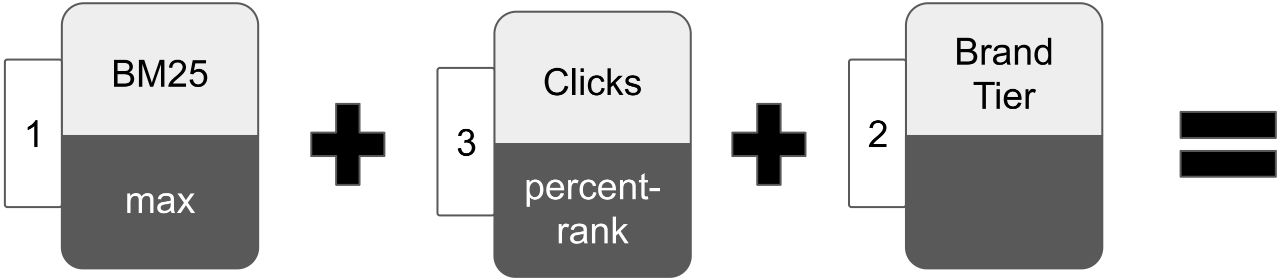


Time to combine

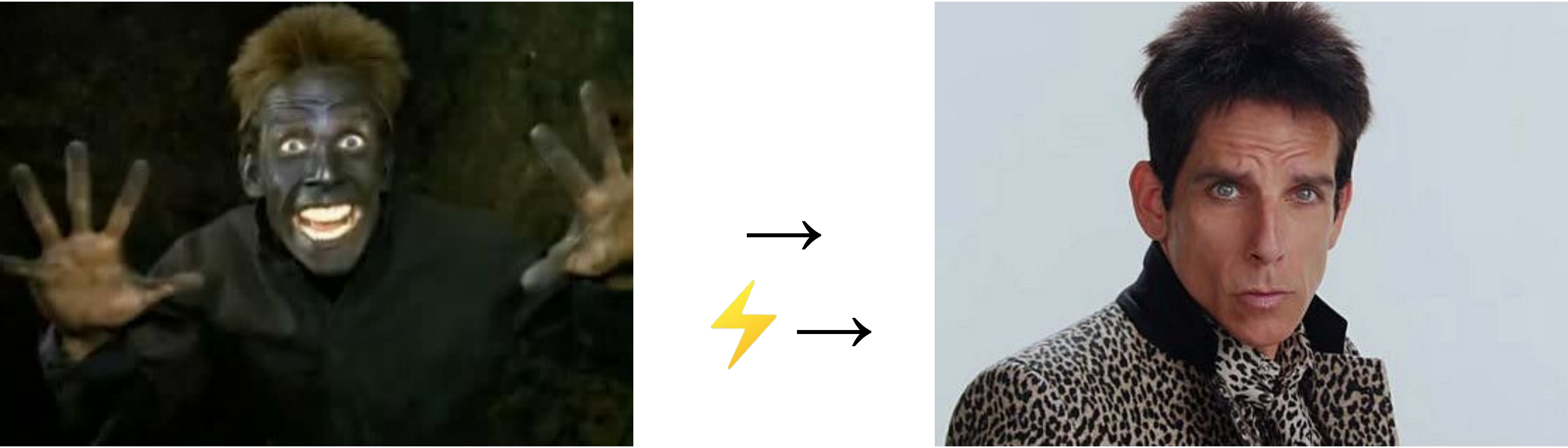
Linear combinations

Individual coefficient (weight) per ranking factor

Sum across weighted ranking factors



Danke



Your existing data may look rough but with some polishing it will become a beautiful ranking feature for your business

@natedayta / @o19s