

ACTIVATE 19

The Growth Marketing Conference

Building A Marketing Castle on a Strong Data Foundation

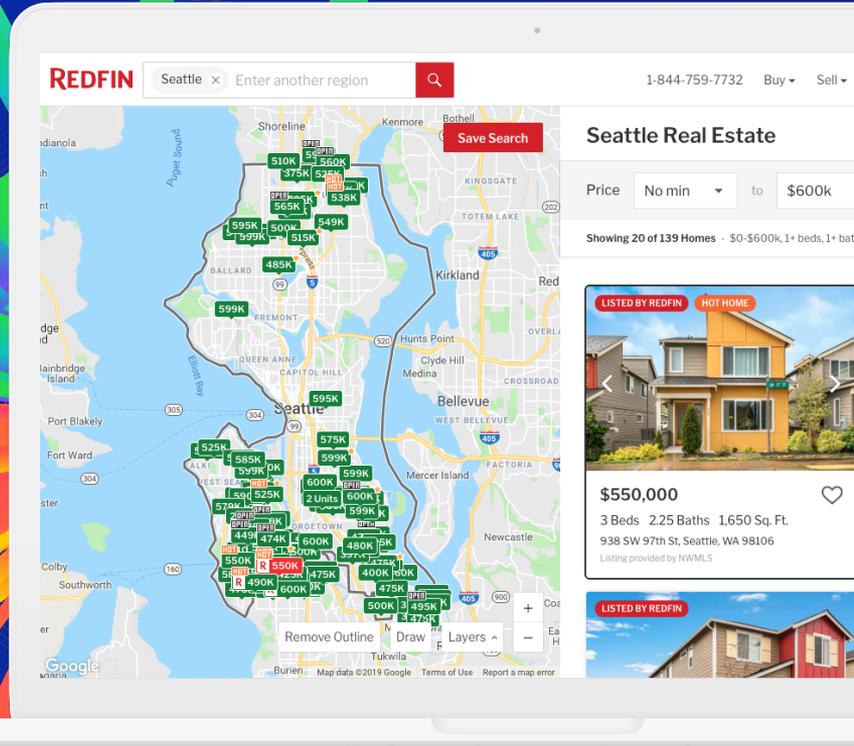
Katie Gillespie - Product Manager, Redfin

Presented by:  *iterable*



Introductions

- Redfin
 - Real estate brokerage
 - Full-service agents
 - Technology-powered
- Me: started in email marketing; transitioned to “technical marketing”
- Led our Iterable integration



Agenda

1. Primary data structures in Iterable
2. When to use each one
3. Redfin's integration process: translating our data into Iterable's format



Why Redfin Switched to Iterable

- Stretching the limits of the old software
- Wrote SQL queries for every email sent
 - Time-consuming to build & test
 - Inconsistent and buggy
 - Hard to hire for
- No big picture view of our campaigns

```
coalesce(a_agt.agent_id, r_agt.agent_id) as agent_id,
case when a_agt.agent_id is not null then 1 else 0 end as use_client_phone

from (
  select
  lgn.,
  case
  when lpr.list_score >= 0.2 then 'marketing-sell'
  else 'marketing-buy'
  end as user_type,
  '1010612' as camp_id,
  zip.zip_code as user_zip_code,
  nullifint as property_id,
  zip.business_market_id as _zip_business_market_id

from ds_internal_38208 lgn

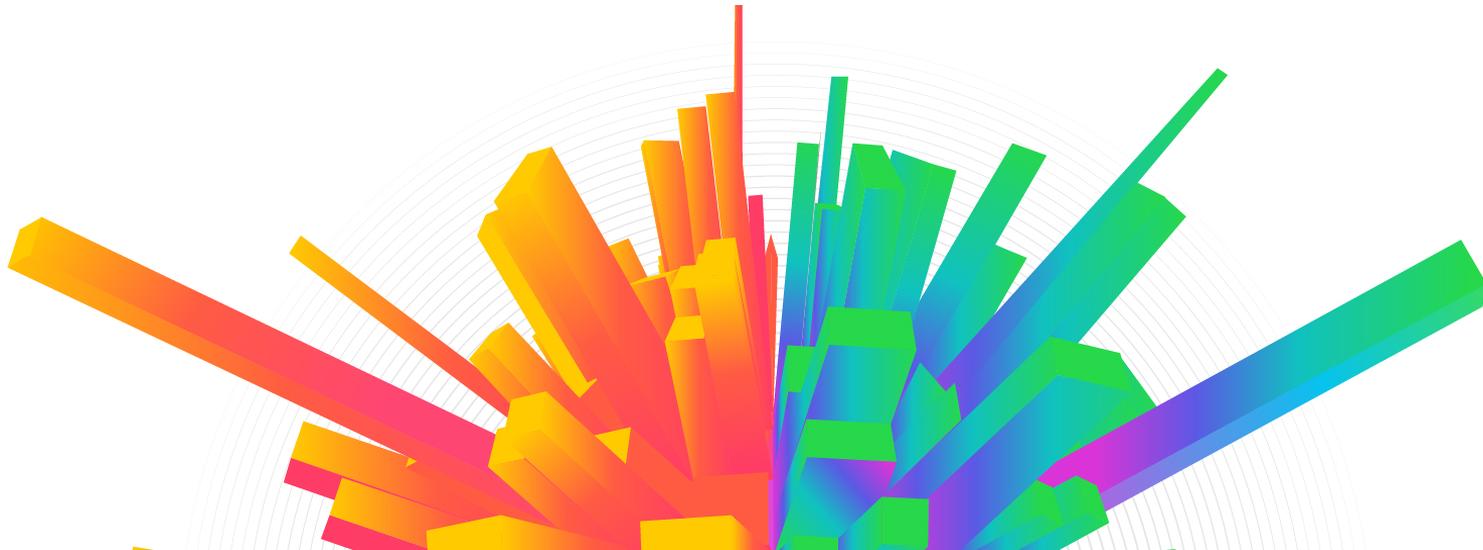
--People with a high lpp score, viewed in the last 28 days, and at least a minimal user score
left join ds_extension_37692 lpp on lpp.login_id = lgn.login_id
and lpp.lpp_rank = '1'
and lpp.lpp_score >= 10
and lpp.lpp_score_time_since_last_view <= 28
and lpp.user_score >= 0.01
left join ds_extension_78401 lpr on lpr.login_id = lgn.login_id

inner join ds_extension_37700 zip on lpp.zip_code = zip.zip_code
inner join ds_extension_37800 srg on srg.service_region_id = zip.service_region_id

where 1=1
and lpp.price_high >= srg.direct_only_threshold
) t

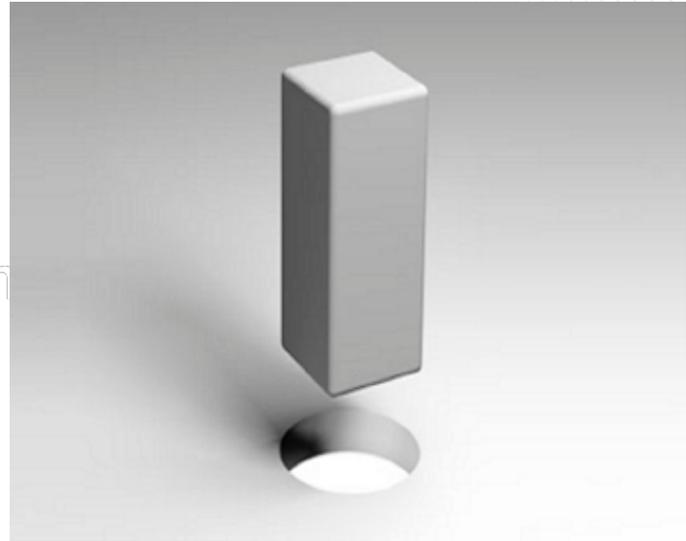
--Agent stuff
left join (
  select
  login_id,
  agent_id,
  relationship_type,
  row_number() over (partition by login_id, relationship_type order by start_date desc) as row_id
from ds_extension_37506
) lar on lar.login_id = t.login_id
and lar.row_id = 1
and relationship_type = case
when t.user_type = 'marketing-sell' then 'Sell'
when t.user_type = 'marketing-buy' then 'Buy'
else 'Buy'
end
left join ds_extension_37500 a_agt on a_agt.agent_id = lar.agent_id
and a_agt.agent_active flag = 'Y'
```

Transitioning to Iterable



Completely New Data Structure

- SQL
 - Used by old software and our internal database
 - Very structured
- NoSQL
 - Used by Iterable
 - Flexible format
 - Blazin' fast
- Needed to reformat our data dramatically to fit new system



SQL

login_id	email_address	first_name	is_active
1	jaiden@gmail.com	Jaiden	TRUE
2	parth@yahoo.com	Luis	FALSE
3	sofia@gmail.com	Sofia	TRUE

NoSQL

jaiden@gmail.com

login_id: 1

first_name: Jaiden

is_active: TRUE

parth@yahoo.com

login_id: 2

first_name: Luis

is_active: FALSE

sofia@gmail.com

login_id: 3

first_name: Sofia

is_active: TRUE

ACTIVATE 19
Growth Marketing Conference

Primary Iterable Data Structures



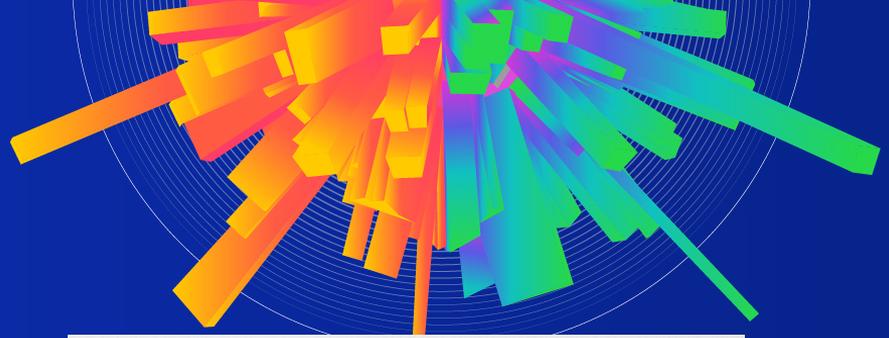
Three Main Lists in Iterable

1. User Profile
2. Custom Events
3. Catalog



The User Profile

- Characteristics about a user
- Have a particular value at any moment in time
- Examples:
 - Registration reason
 - Is active
 - Favorite product type
 - Lifetime value



SQL Table: Logins

login_id	email_address	first_name	is_active
1	jaiden@gmail.com	Jaiden	TRUE
2	parth@yahoo.com	Luis	FALSE
3	sofia@gmail.com	Sofia	TRUE

SQL Table: Logins

login_id	email_address	first_name	is_active
1	jaiden@gmail.com	Jaiden	TRUE
2	parth@yahoo.com	Luis	FALSE
3	sofia@gmail.com	Sofia	TRUE

User Profile

jaiden@gmail.com
login_id: 1
first_name: Jaiden
is_active: TRUE

parth@yahoo.com
login_id: 2
first_name: Luis
is_active: FALSE

sofia@gmail.com
login_id: 3
first_name: Sofia
is_active: TRUE

SQL Table: Logins

login_id	email_address	first_name	is_active
1	jaiden@gmail.com	Jaiden	TRUE
2	parth@yahoo.com	Luis	FALSE
3	sofia@gmail.com	Sofia	TRUE

SQL Table: Claimed Homes

claimed_home_id	login_id	claim_address	claim_date
1	3	1054 SW Briarwood Rd	2019-02-05
2	2	4464 N 43rd St	2019-03-01

User Profile

jaiden@gmail.com
login_id: 1
first_name: Jaiden
is_active: TRUE

parth@yahoo.com
login_id: 2
first_name: Luis
is_active: FALSE

sofia@gmail.com
login_id: 3
first_name: Sofia
is_active: TRUE

SQL Table: Logins

login_id	email_address	first_name	is_active
1	jaiden@gmail.com	Jaiden	TRUE
2	parth@yahoo.com	Luis	FALSE
3	sofia@gmail.com	Sofia	TRUE

SQL Table: Claimed Homes

claimed_home_id	login_id	claim_address	claim_date
1	3	1054 SW Briarwood Rd	2019-02-05
2	2	4464 N 43rd St	2019-03-01

User Profile

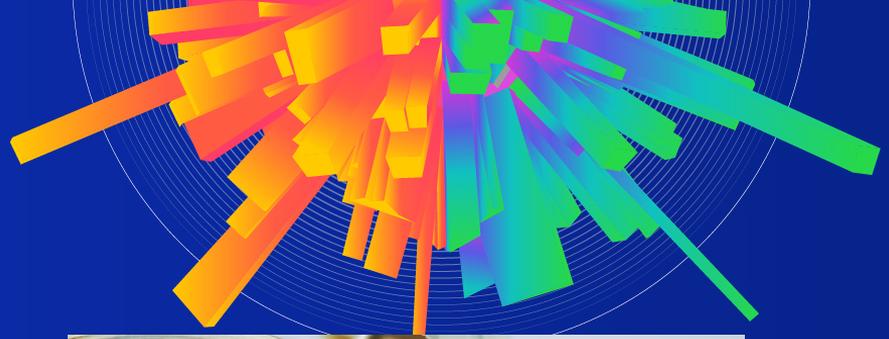
jaiden@gmail.com
login_id: 1
first_name: Jaiden
is_active: TRUE

parth@yahoo.com
login_id: 2
first_name: Luis
is_active: FALSE
claimed_home:
claim_address: 4464 N 43rd St
claim_date: 2019-03-01

sofia@gmail.com
login_id: 3
first_name: Sofia
is_active: TRUE
claimed_home:
claim_address: 1054 SW Briarwood Rd
claim_date: 2019-02-05

Custom Events

- What a user *does*
- Things which happen at a particular moment in time
- Examples:
 - Clicked a button on the site
 - Attended an event
 - Purchased an item



SQL Table: Class Attendees

email_address	class_id	registered_date
mitsuki@gmail.com	1	2019-01-01
conor@hotmail.com	1	2019-01-06
mitsuki@gmail.com	2	2019-02-03

User Profile

mitsuki@gmail.com

login_id: 4
first_name: Mitsuki
is_active: TRUE

conor@hotmail.com

login_id: 5
first_name: Conor
is_active: TRUE

SQL Table: Class Attendees

email_address	class_id	registered_date
mitsuki@gmail.com	1	2019-01-01
conor@hotmail.com	1	2019-01-06
mitsuki@gmail.com	2	2019-02-03

User Profile

mitsuki@gmail.com

login_id: 4
first_name: Mitsuki
is_active: TRUE
class_registered:
class_id: 1
registered_date: 2019-01-01

conor@hotmail.com

login_id: 5
first_name: Conor
is_active: TRUE

SQL Table: Class Attendees

email_address	class_id	registered_date
mitsuki@gmail.com	1	2019-01-01
conor@hotmail.com	1	2019-01-06
mitsuki@gmail.com	2	2019-02-03

Custom Events

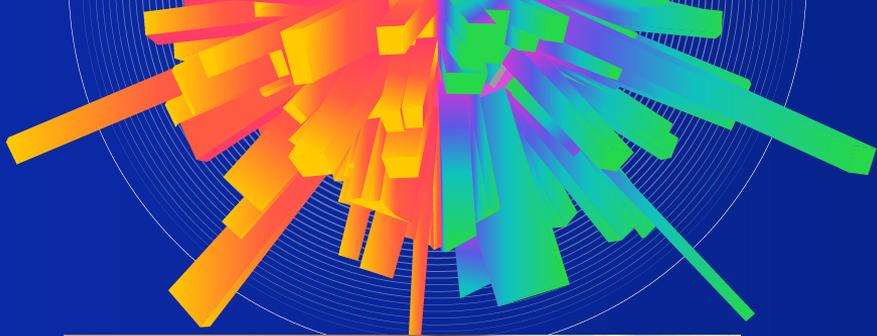
mitsuki@gmail.com
registered_date: 2019-01-01
class_id: 1

conor@hotmail.com
registered_date: 2019-01-06
class_id: 1

mitsuki@gmail.com
registered_date: 2019-02-03
class_id: 2

Catalog

- Enriches other data
- Shared across users
- Examples:
 - Events: location, time
 - Items: prices, colors, urls
 - Agents: names, emails, photos



SQL Table: Class Attendees

email_address	class_id	registered_date
mitsuki@gmail.com	1	2019-01-01
conor@hotmail.com	1	2019-01-06
mitsuki@gmail.com	2	2019-02-03

SQL Table: Classes

class_id	class_date	class_location
1	2019-01-11	Redfin HQ
2	2019-02-08	The Local Grill

Custom Events

mitsuki@gmail.com
registered_date: 2019-01-01
class_id: 1
class_date: 2019-01-11
class_location: Redfin HQ

conor@hotmail.com
registered_date: 2019-01-06
class_id: 1

mitsuki@gmail.com
registered_date: 2019-02-03
class_id: 2

SQL Table: Class Attendees

email_address	class_id	registered_date
mitsuki@gmail.com	1	2019-01-01
conor@hotmail.com	1	2019-01-06
mitsuki@gmail.com	2	2019-02-03

SQL Table: Classes

class_id	class_date	class_location
1	2019-01-11	Redfin HQ
2	2019-02-08	The Local Grill

Custom Events

mitsuki@gmail.com
registered_date: 2019-01-01
class_id: 1
class_date: 2019-01-11
class_location: Redfin HQ

conor@hotmail.com
registered_date: 2019-01-06
class_id: 1

mitsuki@gmail.com
registered_date: 2019-02-03
class_id: 2

SQL Table: Class Attendees

email_address	class_id	registered_date
mitsuki@gmail.com	1	2019-01-01
conor@hotmail.com	1	2019-01-06
mitsuki@gmail.com	2	2019-02-03

SQL Table: Classes

class_id	class_date	class_location
1	2019-01-11	Redfin HQ
2	2019-02-08	The Local Grill

Custom Events

mitsuki@gmail.com
registered_date: 2019-01-01
class_id: 1

conor@hotmail.com
registered_date: 2019-01-06
class_id: 1

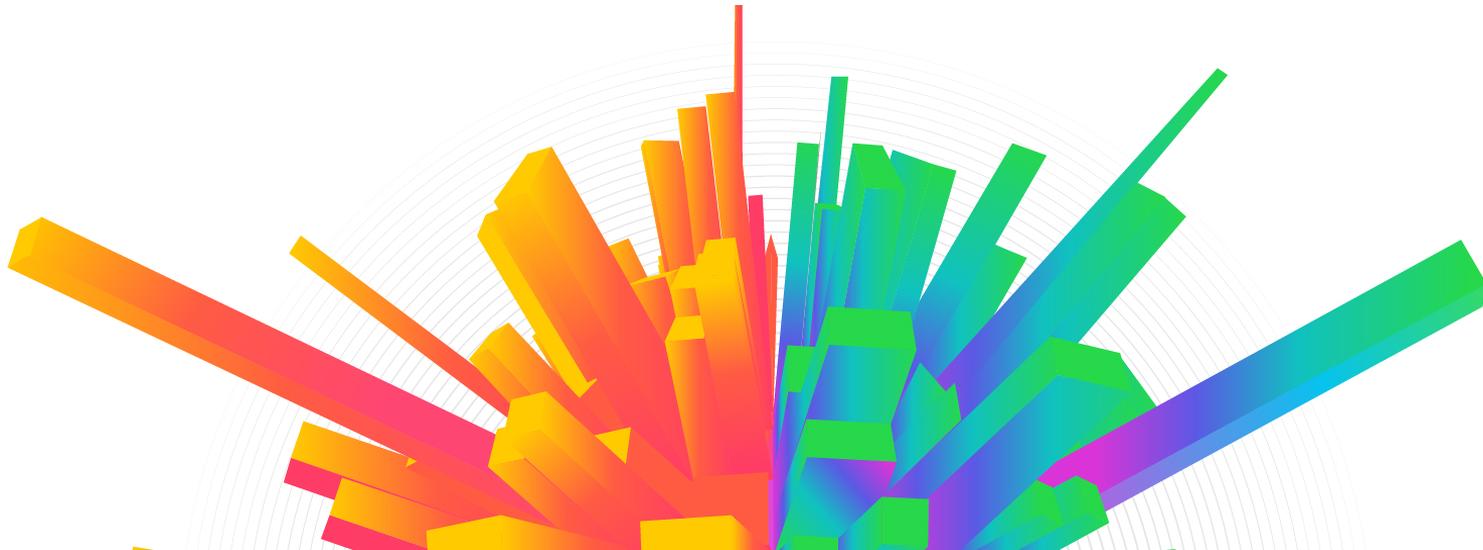
mitsuki@gmail.com
registered_date: 2019-02-03
class_id: 2

Classes Catalog

key: 1
class_date: 2019-01-11
class_location: Redfin HQ

key: 2
class_date: 2019-02-28
class_location: The Local Grill

When to Use Each Kind of List



Use Cases for Each List Type

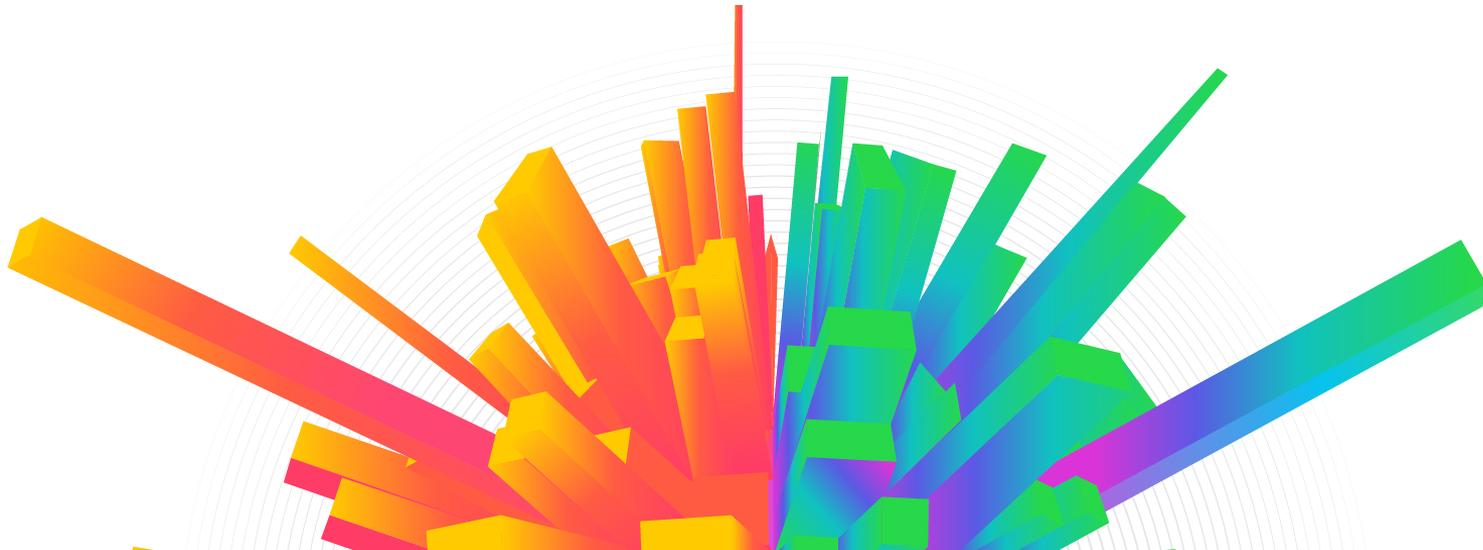
How You Can Use Data	User Profile	Custom Event	Catalog
Segment for matching users			
Trigger an email or workflow	Indirectly, in batches		
Customize templates		Only with the triggering event	

- Can use workflows to get around some of these limitations

List Type for Different Situations

If you want to...	Your best option is....
Send batch campaigns to all users like this	User Profile Field
Change email content based on whether users have done this	User Profile Field
Trigger an email campaign when this event happens	Custom Event
Mainly look at this behavior in aggregate	User Profile Field
Mainly look at this behavior individually	Custom Event
Share a big chunk of data across many users	Catalog
Use data to segment a campaign	Not a Catalog

Notable Examples



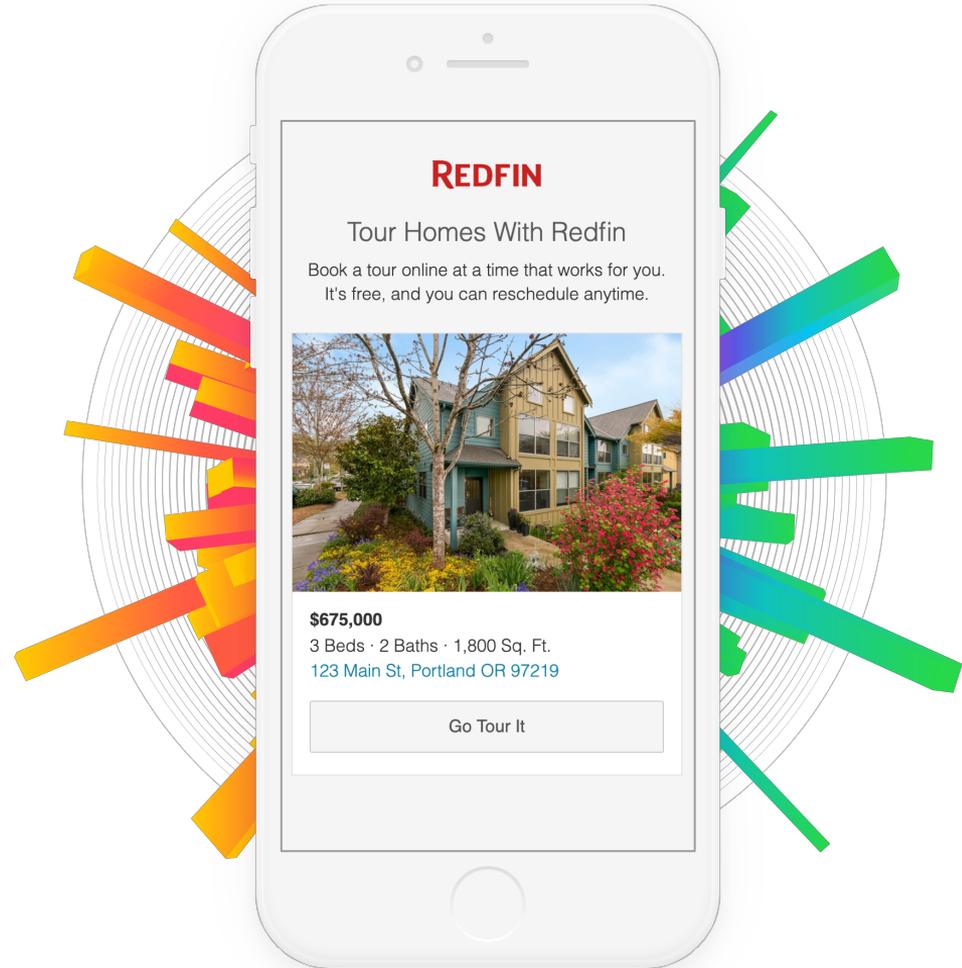
Recommended Agents

- Deciding which Redfin agent to show to users
- In old software:
 - Each email used a different methodology
- In Iterable:
 - Centralized
 - One algorithm on the back end
 - Show agent in User Profile

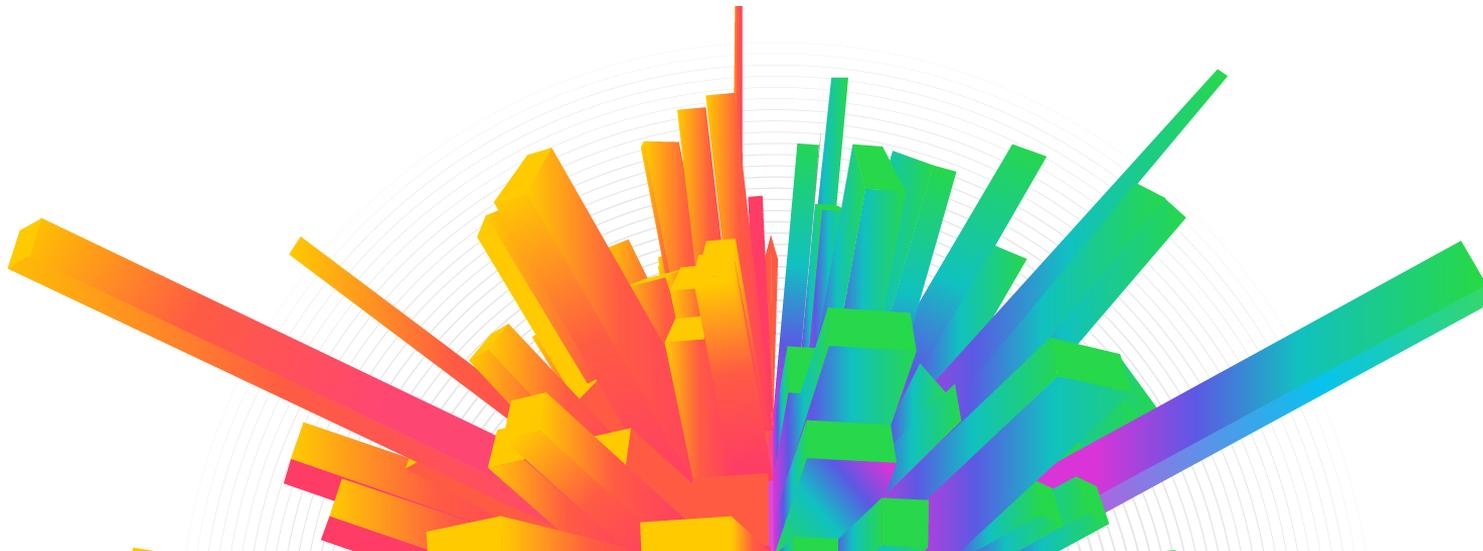


Favorite Listings

- Encourage users to tour their favorite homes
 - Up to 4 per email
- Decided on User Profile
 - Problem with array search
 - Used a “rank” to solve
- Might use custom events
 - Batch up and send a custom event daily



Redfin's Transition Process



Deciding on Data

- Looked at each current query
 - Listed out data points we used
 - Identified how we used them
- Brainstormed “wishlist” data
- Created a list of all the data points needed
- Picked a data type for each

New Registrant

Old Mailing Names

Q32017 New Registrant - Email 5
Q32017 New Registrant - Email 4
Q32017 New Registrant - Email 3
Q32017 New Registrant - Email 2
Q32017 New Registrant - Email 1

New Workflows

S* New Registrant

Putting all 5 into one series. Including Mortgage email and Open house plaintext emails.

Audience:

- Main emails, AND:
 - Trigger: registration date
 - last_interaction_date is null
 - Registration_reason != 'open-house-sign-in'
 - (email 3) OR:
 - Buyside_email_agent_id is not null
 - AND:
 - Sellside_email_agent_id is not null
 - List_score >= 0.1
- Mortgage:
 - expected_buying_market_id in (mortgage list)
 - No custom event for tour email
 - User score >= 0.1 or something
 - Not:
 - Item in deals
 - deal_status like '%offer%'
 - Deal_status_is_buyside_status_flag = true
- Open House Plaintext:
 - Contact_post_open_house_flag = true
 - Registration_reason = 'open-house-sign-in'
 - Open_house_date = registration_date
 - Host_redfin_agent_id, listing_agent_id, buyside_email_agent_id,
- Awareness survey:
 - Valid_for_surveys_flag = false
 - Lpp_above_redfin_threshold_flag = true

Uploading Data

- Wrote a giant query
 - One for the user profile
 - One for each custom event type
- User Profile: nightly batch process
 - Straightforward & reliable
 - Downside: slow
- Custom Events: hourly process
 - Find all events which happened since the last time the process ran
- Some engineering teams send custom events to us directly

Where We Are Now: Efficiency & Growth

- Team is much more efficient
 - Faster to create, test, & launch
- Targeting specific users easily
- Experiencing fewer bugs
 - More reliable data
- Reaching users we'd missed
- Integrating directly with Engineering teams

+124%

Year-over-year growth
in attributed contacts
6 months after integrating



User Profile
Qualities & Descriptors



Custom Events
Moments in Time



Catalog
Shared Information



ACTIVATE 19

The Growth Marketing Conference

Q&A

If you have other questions:

- katie.gillespie@redfin.com
- [linkedin.com/in/kathringillespie](https://www.linkedin.com/in/kathringillespie)

