Collaborative Recommendations

ECG Finding Science (FiSci) Team

June 29, 2016

Welcome to Kijiji!....and

dba

kijiji™
Canada's largest classifieds site

mobile.de

Gumtree

Close5

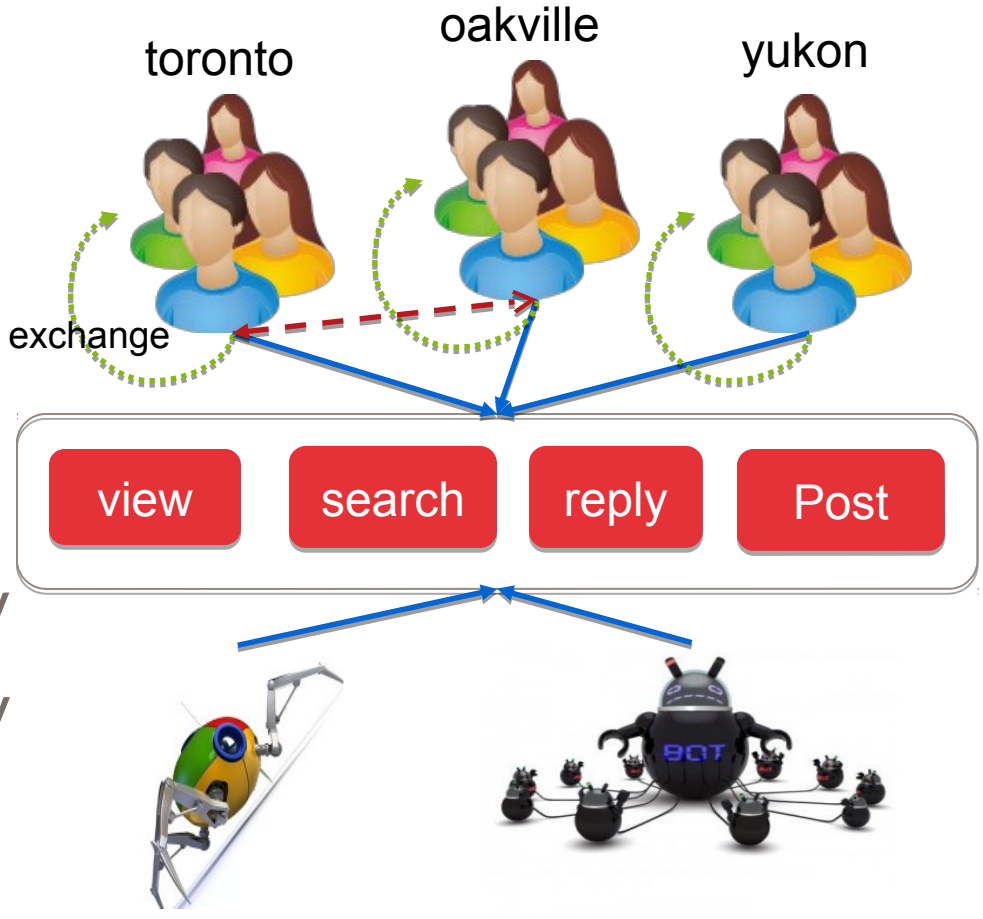
MARKTPLAATS.NL

Format

- Introductions
- Framing the problem and thinking through it.
- Short break
- Taking the solution to spark (the code)
- Other things involved in the overall system

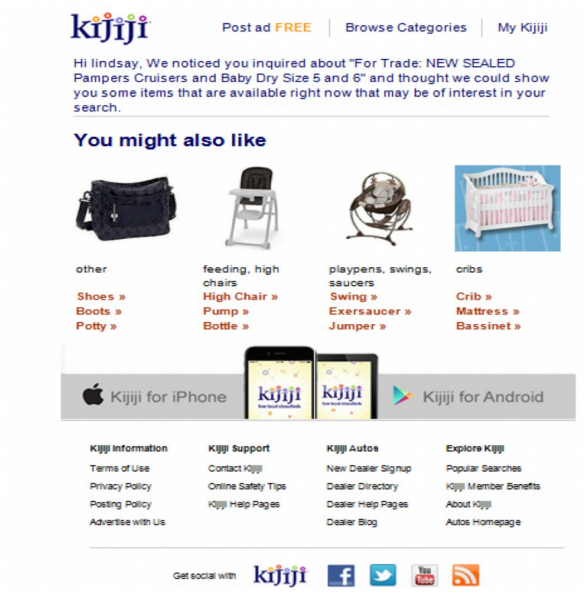
What all our markets have in common…

- Local Community driven
- Highly anonymous interactions / low touch
- Diverse users with ever changing/different needs
- Locality sensitive inventory
- Rapidly changing inventory
- Non-human traffic

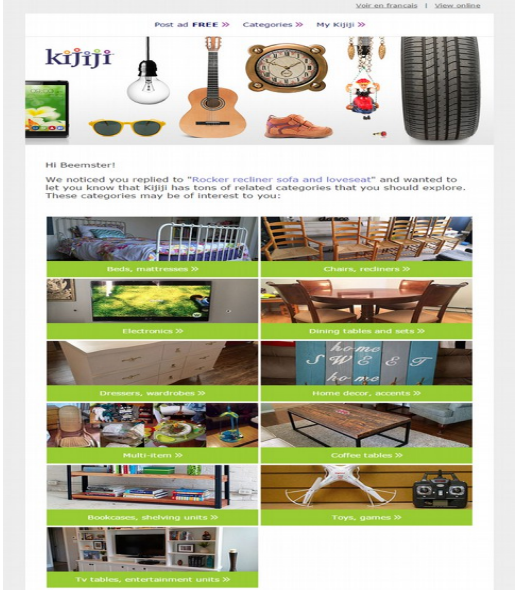


Common Goal

- Enrich the current *offsite re-engagement* experience, driving users back to our platform to *discover* new content and/or help them *succeed*.



Live item level similar / related per category



Idea

- Recommend items for re-engagement to buyers that *reply* to an item within a given threshold. Use *reply* as the *trigger* signal. Recommendations based off the item replied to.
  - Measure the overall open rate and success of the version of those emails against the baseline
  - Measure the session activity of those users arriving from that email against the baseline and measure the conversion rate on specific items clicked on from the email.
  - Triggered on personalized action, but can we utilize collaborative activity of other users on the site/app?

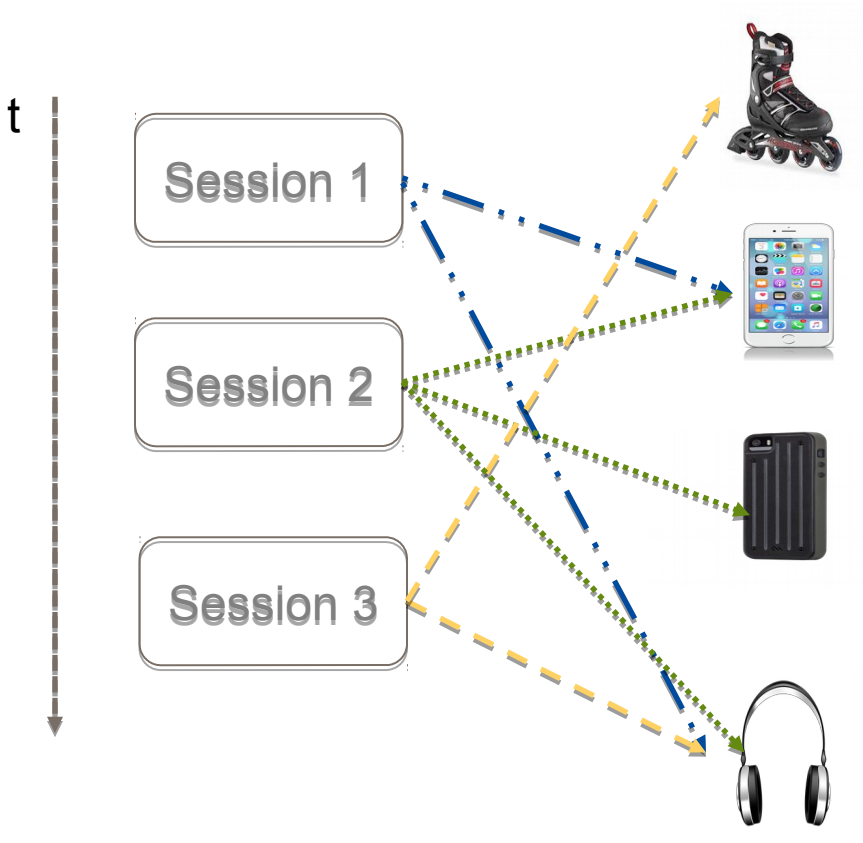
FiSci recommendations

- First pass based on a normalized *co-visitation* algorithm to capture the *collaborative* aspect
- Aiming to capture more '*similarity*' than '*related' but capable of* tuning per category and category trees and other dimensions
- Constraints in place
- Be able to generate recommendations at scale multiple times, quickly and reliably.

Recommendation Techniques

- User / Item Based (Collaborative Filtering)
  - Predictive *model / ML / ALS Scaling can be tough and data very sparse. Doesn't work if you have no underlying STRONG linear latent factors. i.e no strong primary source(s) of variance.*
- Collaborative in general
  - Recommendations determined by *actions* of *entities* within the domain
- Content / Rule Based
  - Intersecting item feature space with user feature space
  - Merchandising / Decision Trees
- Who cares based ☺
  - Almost all end up being hyrbids anyway because most don't perform / can't be tuned optimally on their own

Covisitation



Co-pairs

Recos

Covisitation Memory-Based Algorithm For Recommendation

This memory-based algorithm is simply defined as:-

For a given time period (usually 24 to 48 hours) we count for each pair of adIds (t_i, t_j) in one session how often they were co-visited across all other sessions. Denoting this co-visitation count by c_{ij}, we define the related score of item t_j to base item t_i as:

$$r(t_i, t_j) = \frac{c_{ij}}{f(v_i, v_j)}$$

where c_i and c_j are the total occurrence counts across all sessions for item t_i and t_j, respectively. $f(v_i, v_j)$ is a normalized function that takes the "global popularity" of both the seed item and the candidate item into account.

[1] Davidson, James, et al. "The YouTube video recommendation system. "*Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010.

[2] Carbone, Paris, and Vladimir Vlassov. "Auto-Scoring of Personalised News in the Real-Time Web: Challenges, Overview and Evaluation of the State-of-the-Art Solutions." *Cloud and Autonomic Computing (ICCAC), 2015 International Conference on*. IEEE, 2015.

[3] Das, Abhinandan S., et al. "Google news personalization: scalable online collaborative filtering." *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007.

[4] Bates, K.M., Paas, J., Wang, B., Xu., B. and Yousefi, P., Recommender System for on-line articles and documents, *Google Patents (http://www.google.com/patents/US20090300547)*, 2009, (US Patent App. 12/360, 516).

(This approach has already proved useful for Google News aggregation and for Youtube Collaborative Video Recommendation)

[5] Malik, Z. K., & Fyfe, C. (2012). Review of Web Personalization. *Journal of Emerging Technologies in Web Intelligence* , *4*(3), 285-296.

[6] Malik, Z. K., & Connolly, T. M. (2012). A new personalized approach in affiliate marketing. *e-society*, 235.

[7] Malik, Zeeshan Khawar, Colin Fyfe, and Malcolm Crowe. "Priority recommendation system in an affiliate network." *Journal of Emerging Technologies in Web Intelligence* 5.3 (2013): 222-229.

Computed Symmetrical Covisitation Matrix

ADID / ADID	A_1	A_2	A_3	A_4	A_5	.	.	.	.	A_N
A_1	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	.	.	.	.	C_{1N}
A_2	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	.	.	.	.	C_{2N}
A_3	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}	.	.	.	.	C_{3N}
A_4	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}	.	.	.	.	C_{4N}
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
A_N	C_{N1}	C_{N2}	C_{N3}	C_{N4}	C_{N5}	.	.	.	.	C_{NN}

Primary Reasons behind Selecting Covisitation Memory Based Algorithm

1. Simple and quickest
2. Scalable
3. Contains Efficient Collaborative Personalization Abilities
4. Can be used as a Ranker (for Ad Items) for various other computational purposes.
5. Can be easily integrated with other scalable and efficient Model Based Approaches
6. Maintain Strong Similarity Between Ad Items

Step one – Computing Covisits

Step 2 – Computing Covisits

Session View Pairs

session_id	test_id	rec_id
s1	1	2
s1	2	1
s2	1	2
s2	2	1
s4	3	2
s4	2	3

```
select test_id, rec_id, count(1)
from view_pairs
groupby test_id, rec_id
```

Covisit Triples

test_id	rec_id	covisits
1	2	2
2	1	2
2	3	1
3	2	1

Recos = group by test_id

1 → { 2 }

2 → { 1, 3 }

3 → { 2 }

Step 3 – Can we do better?

- **Symmetric** Relation
  - (a,c) ⇔ (c,a)
  - exploiting symmetry can HALVE your datasize.
  - exploiting symmetry can halve the aggregated key space
  - BIG win with BIG data
- **NOT Transitive**
  - (a,c) & (c,b) !-> (a,b)

- Work with this instead

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1q} \\ a_{21} & a_{22} & & a_{2q} \\ \vdots & & \ddots & \vdots \\ a_{q1} & a_{q2} & \cdots & a_{qq} \end{bmatrix}$$

Preview

- We will leave this out for time / simplicity, but the idea…*(given item id's with a strict partial ordering)*

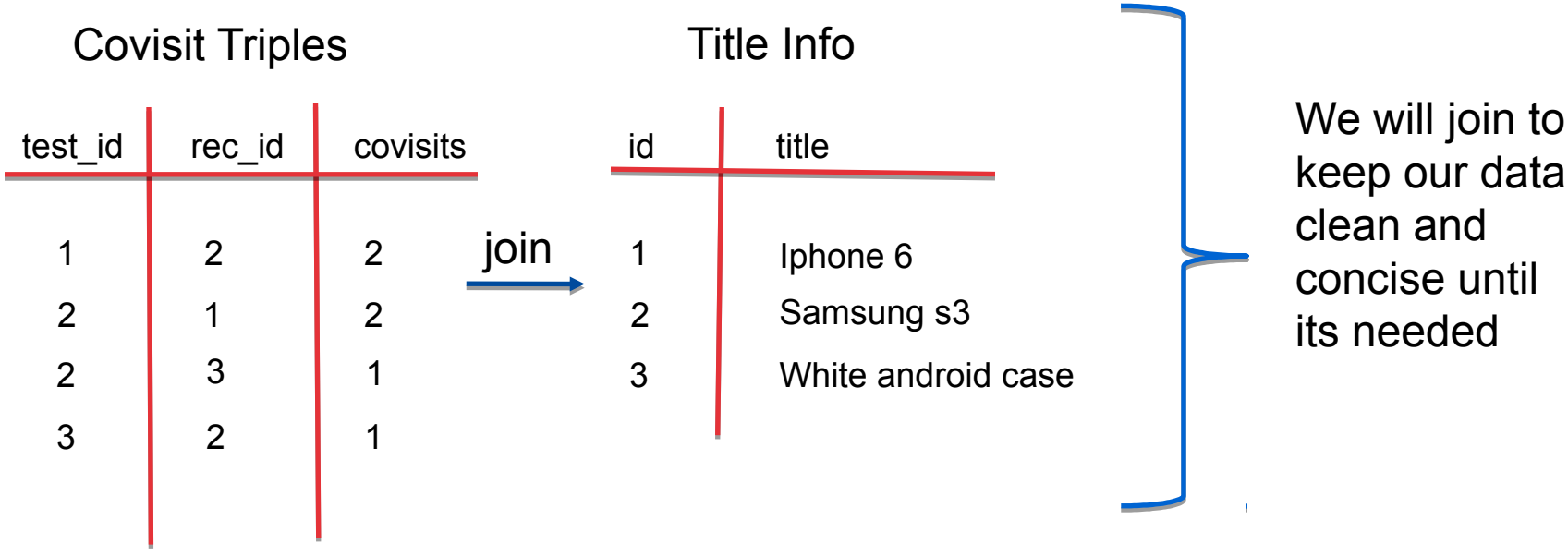
Will require double reduction in the final step. But still more efficient

What about ***f(v_i, v_j)*** ??

- Right now our recommendations for a given item are just an unordered set of covisited items
- Probably not enough given whatever your objective / requirements are.

- E.g
  - *f(v_i, v_j) = Cosine similarity of title v_i with title_vj*
  - *Assumes you carried the title along as a column of each item. problem?*
  - *Score of a given ad is then c_ij * f(v_i,v_j)*
  - *What would this do?*

Bringing in Score



Bringing in Score

Covisits w/ title

test_id	test_t	rec_t	rec_id	covisits
1	t1	t2	2	2
2	t2	t1	1	2
2	t3	t2	3	1
3	t3	t3	2	1

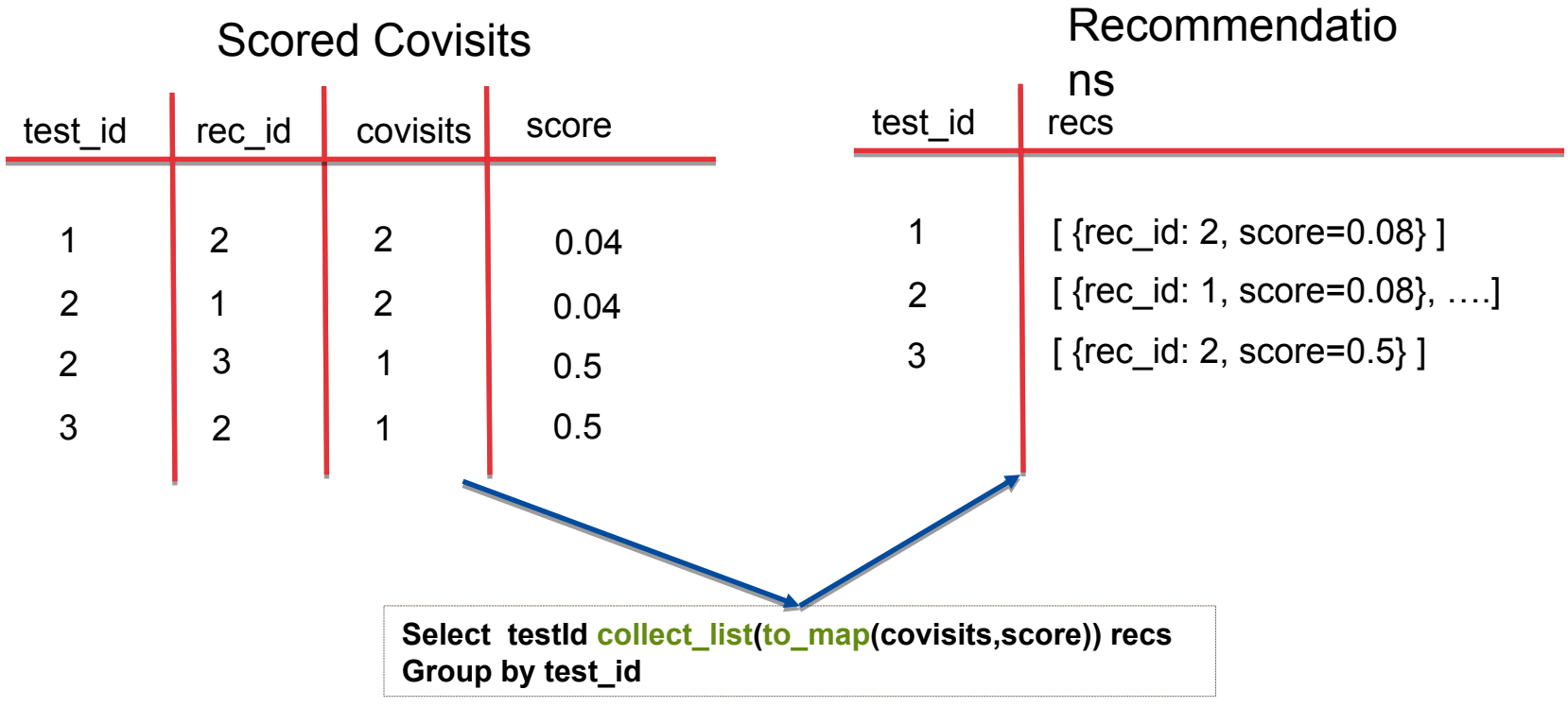
Select test_id, rec_id, covisits * coss(test_t,rec_t) score

Scored Covisits

test_id	rec_id	covisits	score
1	2	2	0.08
2	1	2	0.08
2	3	1	0.50
3	2	1	0.50

Just an example. But what's wrong here?

Final Table of Recommendations



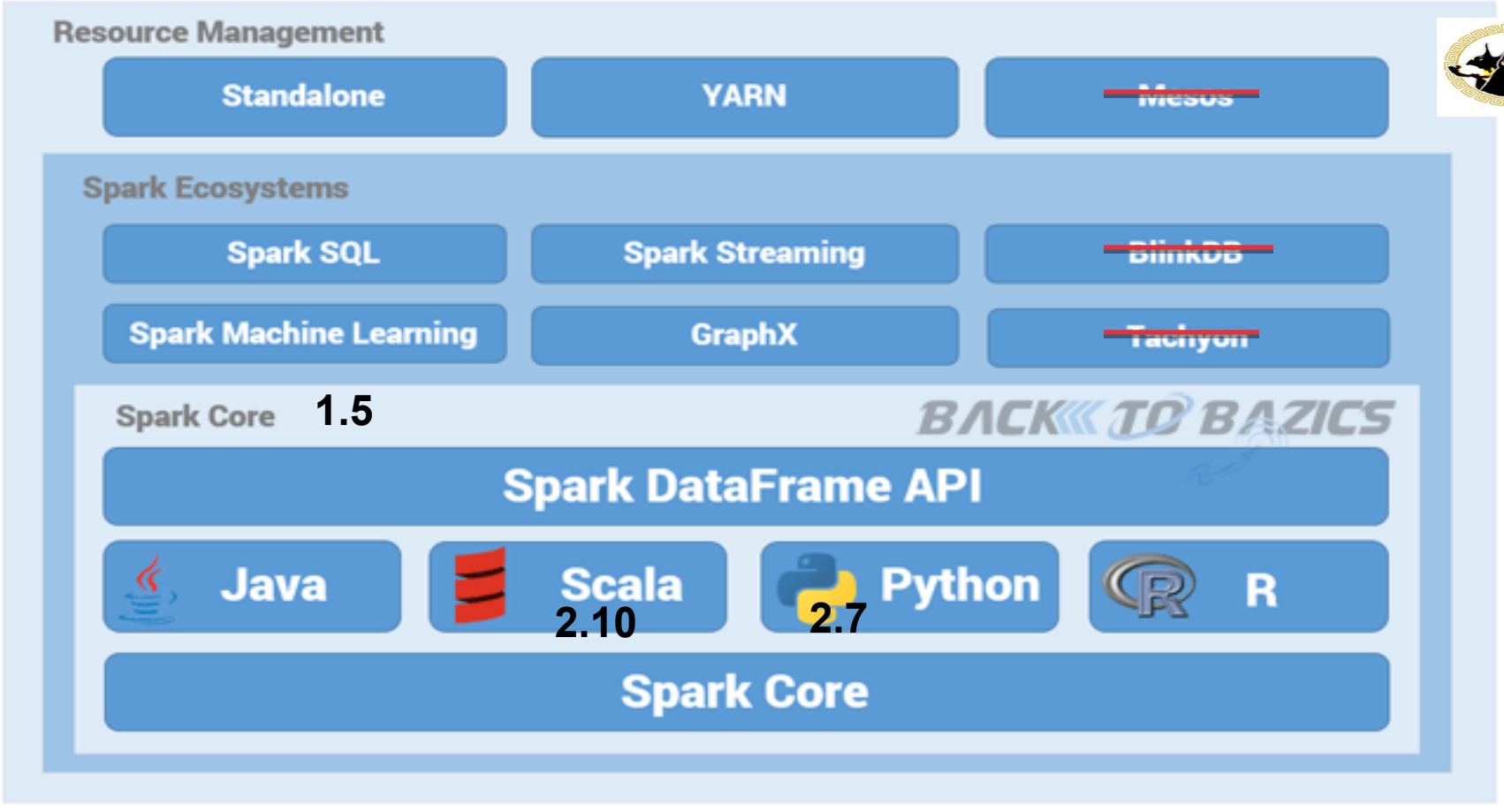
Questions and Break

Taking Covisitation to Spark

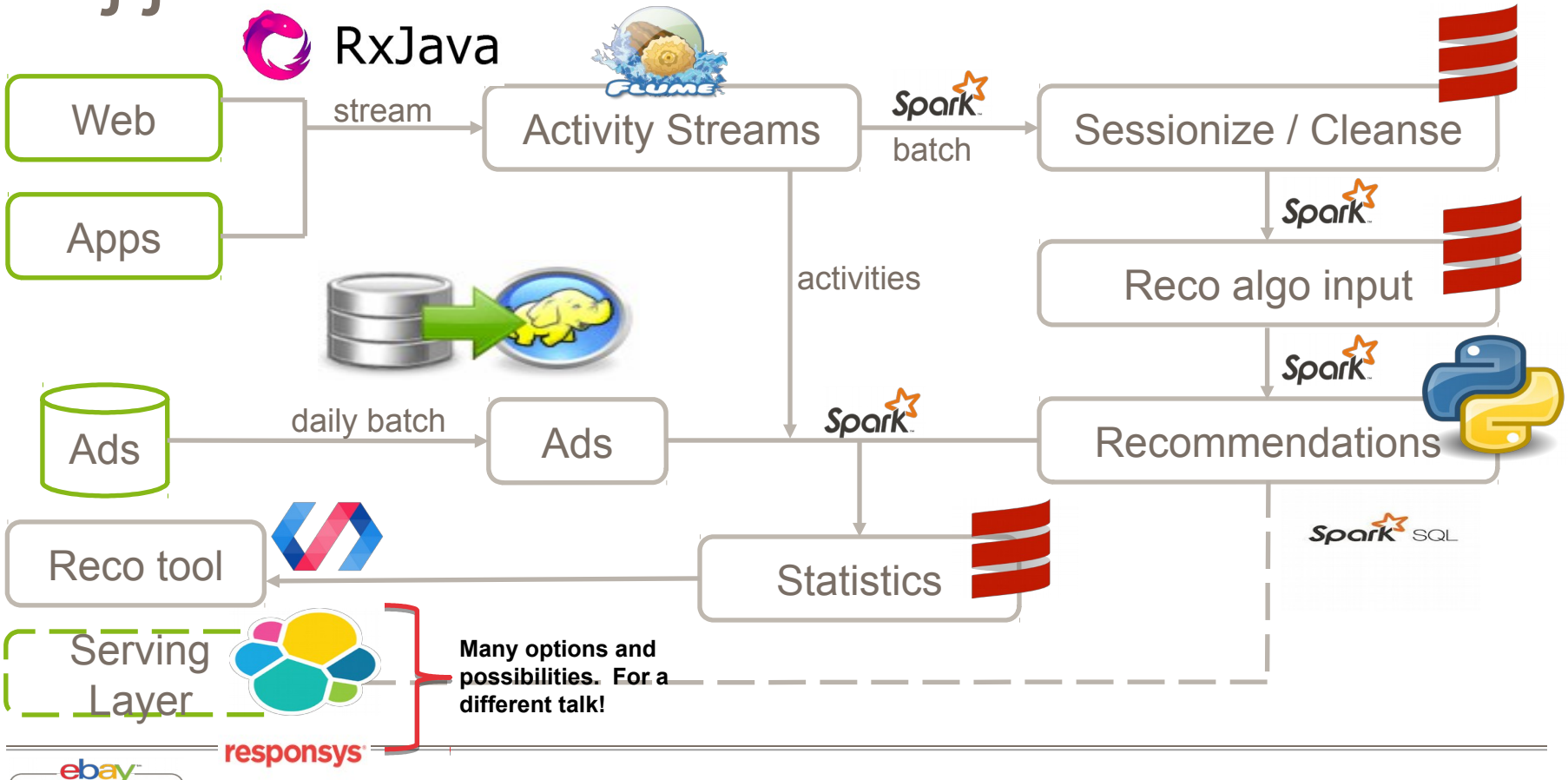
Where the tires meet the road

Our Hadoop World

cloudera®



Kijiji + FiSci recommendations



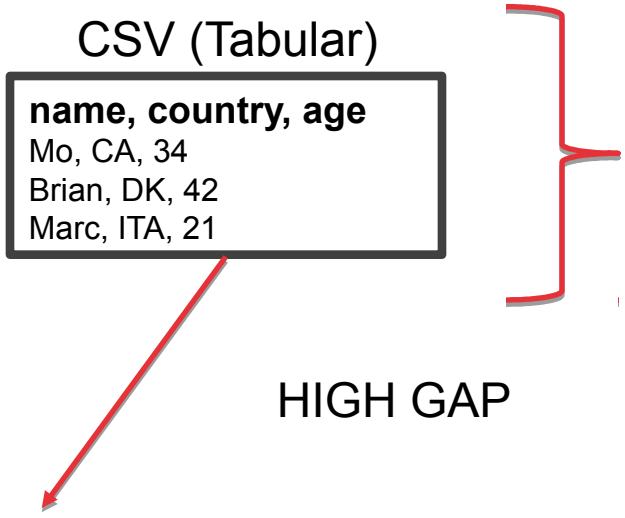
Where did python come from?

- The first project for the lead developer and Python was the most familiar language.
  - Quicker time to delivery
  - Didn't want to add extra complexity / risk to project
  - Easy to port later (We will most likely…)
  - Didn't see any red flags at the time. But of course we found some later ☺

Why SparkSQL / Dataframes?

- Lower Representational Gap
  - Less prone to error staying in our initial reasoning frame. DSL matches 1-1 (for most parts)
- Performance / Complexity
  - Dataframes (in many cases) can give you a major boost in performance for a few reasons…
  - In PySpark, we get less python interpretation and closer direct bytecode execution

Representational Gap (RDD)




```
people_rdd = sc.textFile(namesCsv).map(lambda line: line.split(","))
people_of_country = people_rdd.keyBy(lambda r: r[1]).mapValues(lambda v : v[0])
```

Representational Gap (Dataframe)

A couple lines of boiler plate in the application. But results in LOW gap for rest of the important area

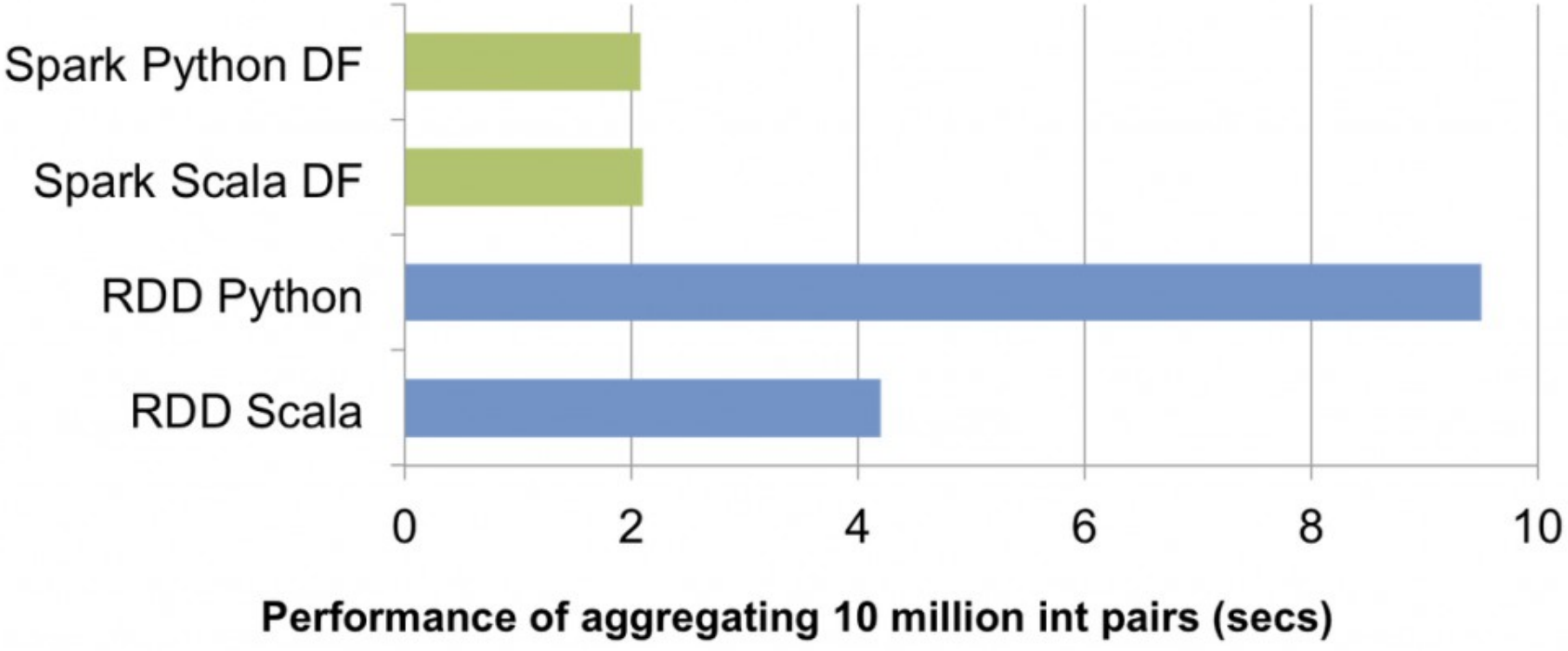
CSV (Tabular)

name, country, age
Mo, CA, 34
Brian, DK, 42
Marc, ITA, 21

```
sqlContext = SQLContext(sc)
people_df = df = sqlContext.read.format('com.databricks.spark.csv') \
      .options(header='true', inferschema='true') \
      .load('names.csv')

people_by_country_df = df.groupBy('country').agg(collect_list('name').alias('names'))

OR

people_df.registerTempTable("people")
people_df = sqlcontext.sql('SELECT country, collect_list(name) names FROM people GROUP BY name')
```

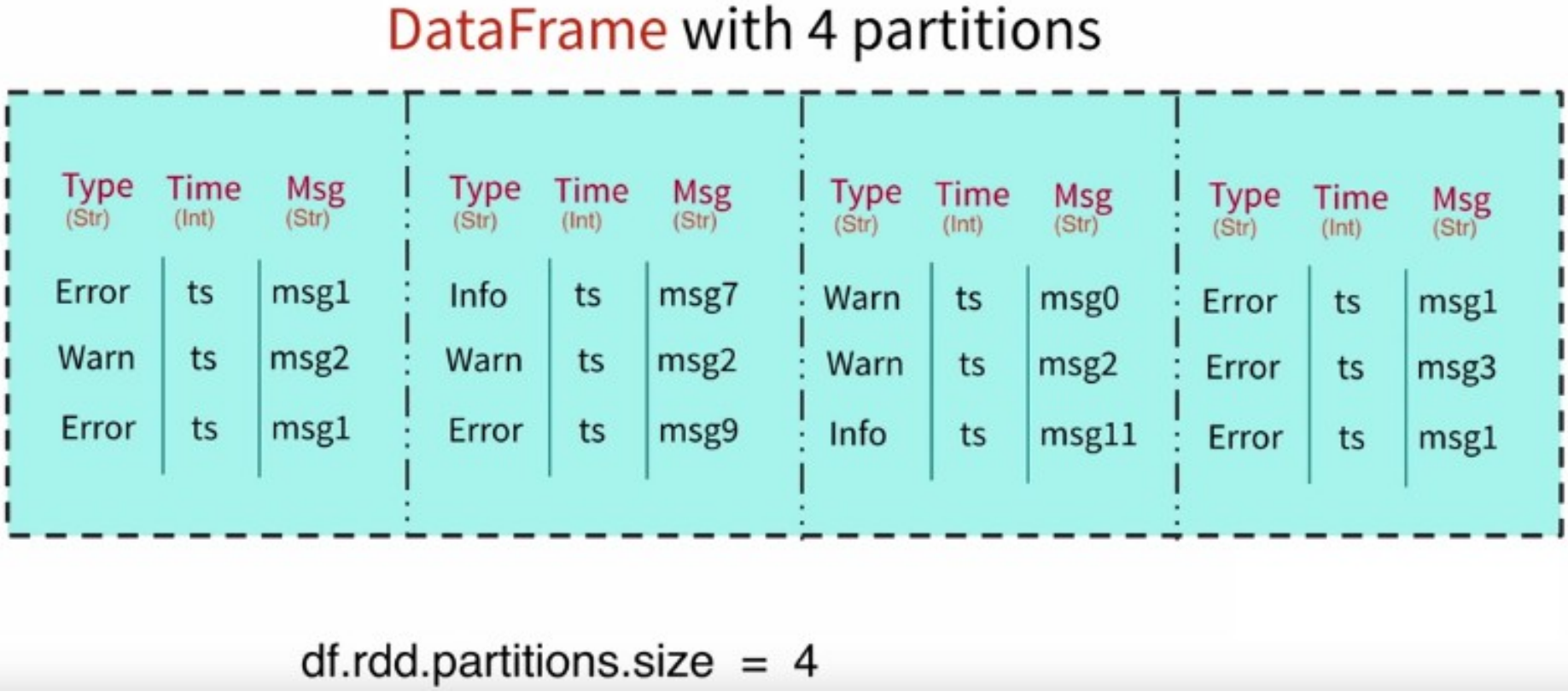
Performance

The infamous benchmark…not always the case. But usually



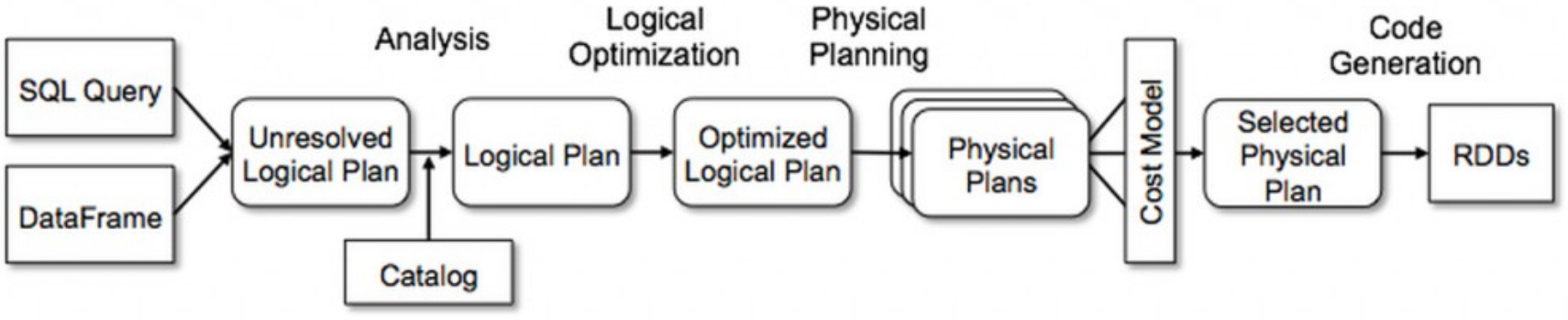
Performance

compress the column value based on the type defined in the column during cache()

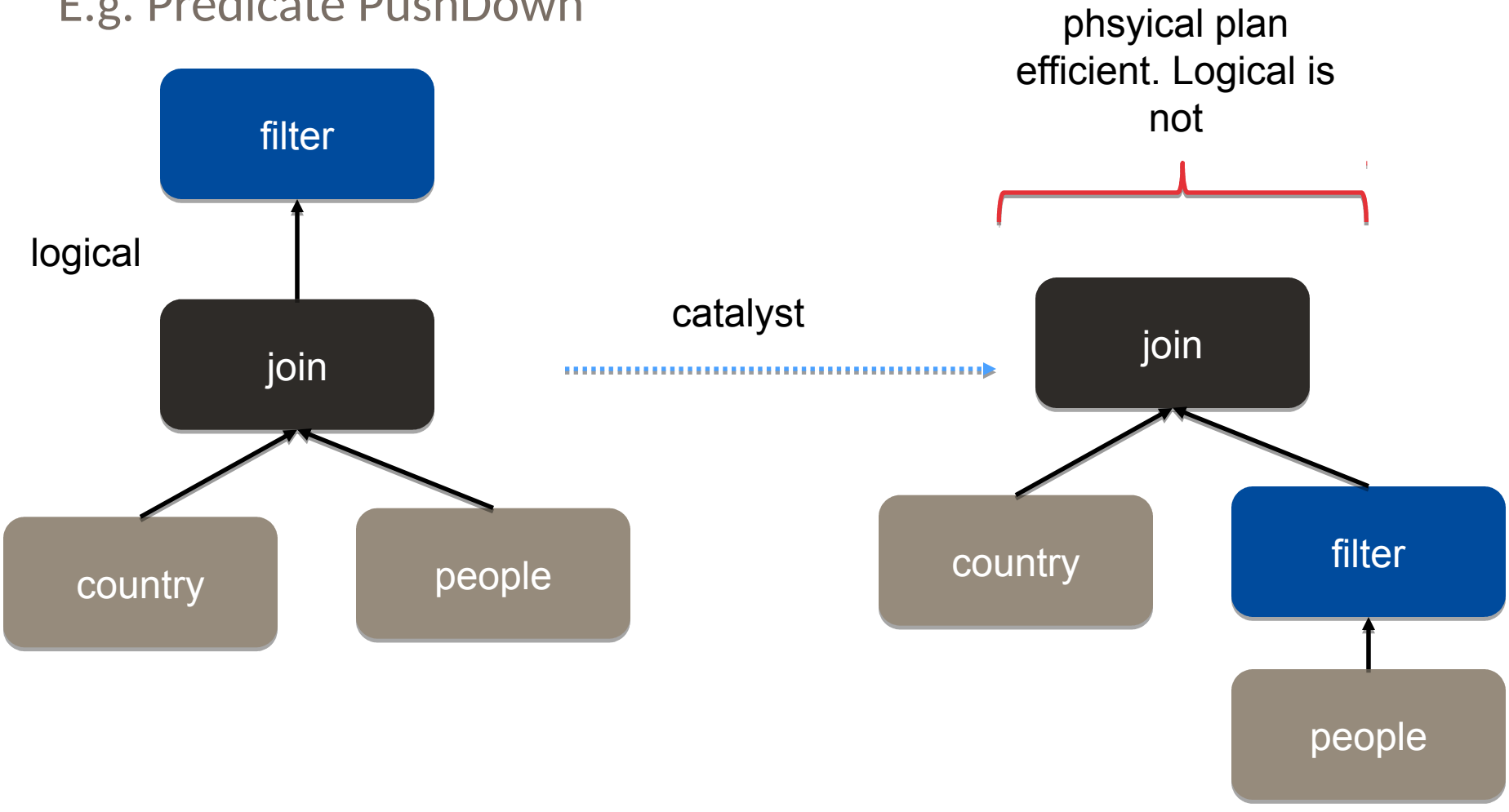


Performance (The Catalyst Optimizer)

spark.sql.shuffle.partitions: 200 ? Huh?



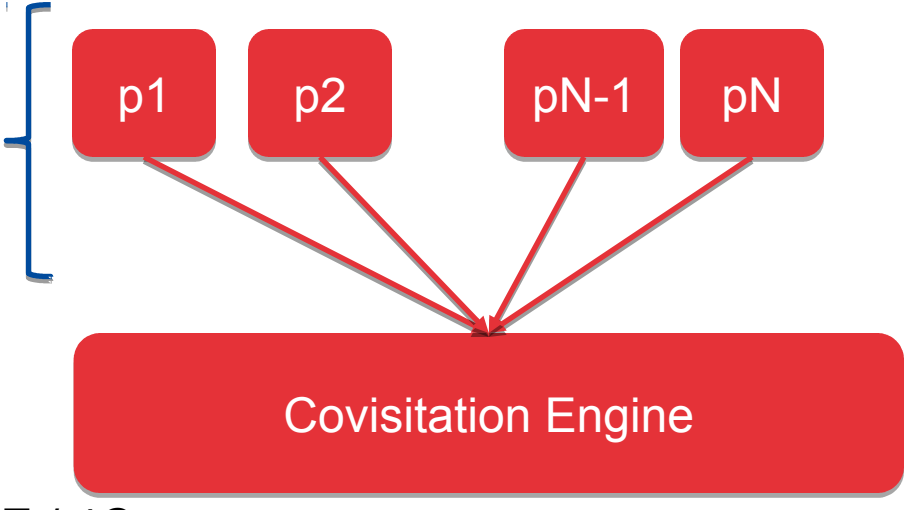
E.g. Predicate PushDown



Starting at input

- Goal
  - A Flexible, compact format that can be easily extended and run through different algorithms / systems
  - Capture all view activities from the last 48 hours
  - Likely will be used for real time case if you want to extend with streaming
  - Difference in performance via Parquet? ***None***. Row oriented algorithm in our case. If projections of the input source are required however, then this would indeed be much less data read for nothing.

About 40 Million Records
20GB



8G / 50E / 4C

Starting At input

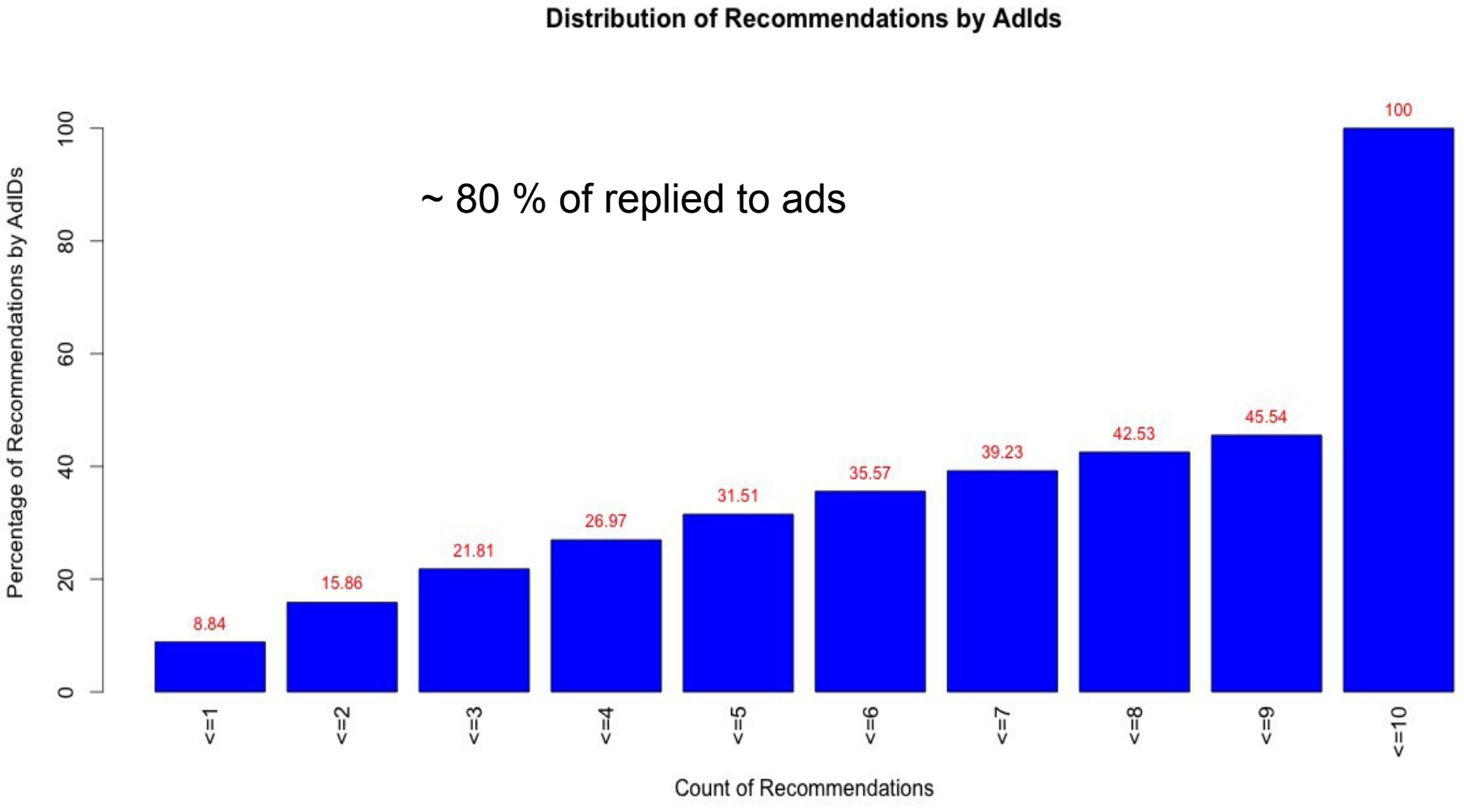
Magic of Cleansing / Sessionizing

```
case class View(sessionId: String,
                adId: String,
                topAd: Boolean,
                hasImages: Boolean,
                l1Category: Int,
                leafCategory: Int,
                l1Location: Int,
                leafLocation: Int,
                userId: String,
                platform: String,
                creationDate: Long,
                lastEditDate: Long,
                viewTime: Long)
```


```
import com.databricks.spark.avro._

val viewDF = viewRDD.toDF()

sqlContext.setConf("spark.sql.avro.compression.codec", "snappy")

viewDF.write.avro(outputPath)
```

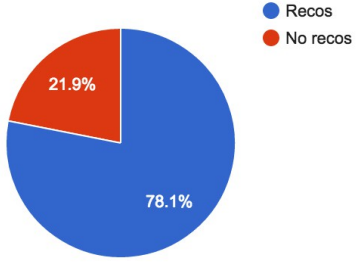
Covisitation (The Code/Tests. Not a lot of it!)

Statistics / Judgements

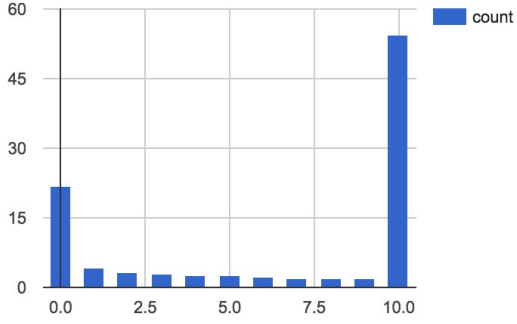

Distribution of Recommendations by AdIds
~ 80 % of replied to ads
Percentage of Recommendations by AdIDs
0
20
40
60
80
100
8.84
15.86
21.81
26.97
31.51
35.57
39.23
42.53
45.54
100
<=1
<=2
<=3
<=4
<=5
<=6
<=7
<=8
<=9
<=10
Count of Recommendations

Selected category: BUY AND SELL

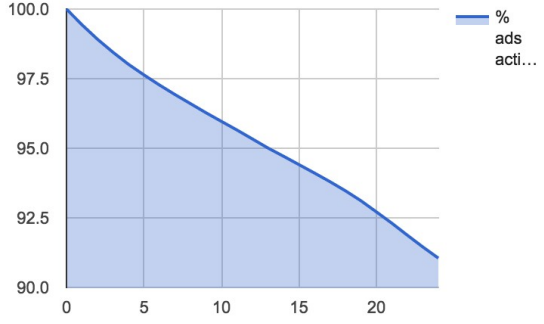
Reply coverage



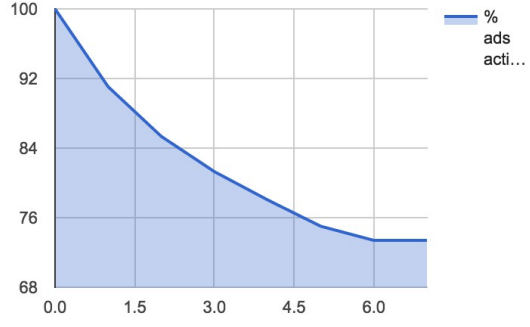
Distribution of #recos/reply



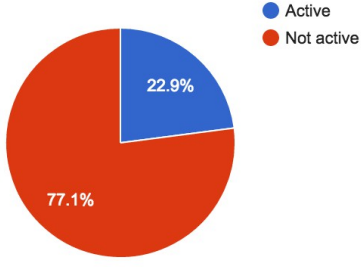
Recommendation mortality (1 day)



Recommendation mortality (1 week)

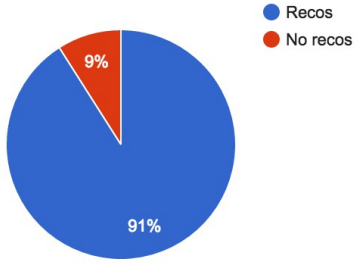


Recommendations still active

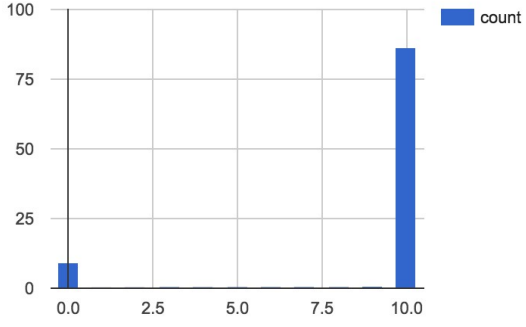


Selected category: USED CARS & TRUCKS

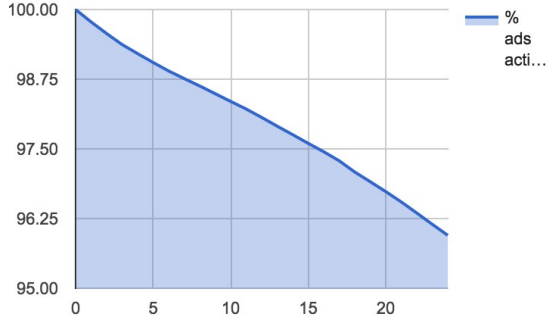
Reply coverage



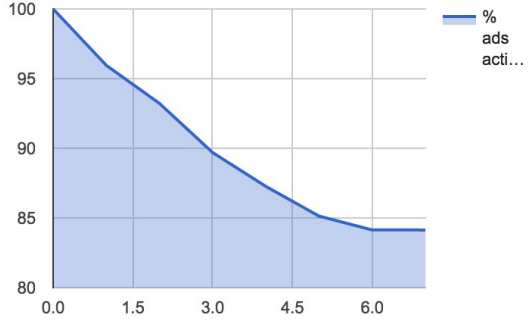
Distribution of #recos/reply



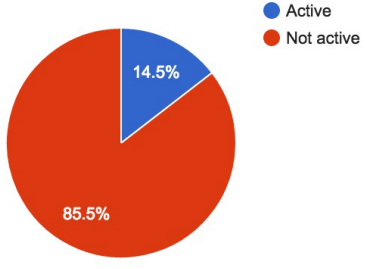
Recommendation mortality (1 day)



Recommendation mortality (1 week)



Recommendations still active



Gauging algorithm ‘quality’

Human judgment tool

Reply to: vintage Fender Super Reverb, tube amp

Time: 2016-04-18 06:58:08.346+0000 (6 days ago)

Category: amps, pedals

Location: Mississauga / Peel Region

Posted: 8 months ago (2015-09-08T01:46:32+0000)

State: ACTIVE

Description: Fender Super Reverb, Silverface from the '70's Ultralinear model with master volume no pedal for the vibrato but everything works, including the...

Number of recommendations: 10

SHOW

Reco: Fender Telecaster - MIJ 1984-1987

Score: 0.021

Category: guitars different category

Location: Markham / York Region different location

Posted: 8 months ago (2015-08-20T19:07:51+0000)

State: ACTIVE

Description: Made at thge Fuji-Jen Plant during the years of 1984 - 1987. Capacitor upgraged to an Sprauge .047 Micro Farad Cap. Upgraded pots. Grover tuners. ...

Recommendation metadata: {"adIdFreq":"3","coVisit":"1","recIDFreq":"16"}

Reco: Fender Pro Reverb 1975 Silverface

Score: 0.167

Category: amps, pedals

Reply to: 2010 Honda CR-V LX DVD

Time: 2016-04-18 14:13:31.433+0000 (6 days ago)

Category: used cars & trucks

Location: Ottawa

Posted: a year ago (2015-02-17T07:24:18+0000)

State: ACTIVE

Description: <div sty...

Number of recommendations: 10

SHOW

Reco: 2012 Honda CR-V LX 5 SPD at 2WD (2)

Score: 0.063

Category: used cars & trucks

Location: Ottawa

Posted: 23 days ago (2016-04-01T19:21:19+0000)

State: ACTIVE

Description: <div sty...

Recommendation metadata: {"adIdFreq":"8","coVisit":"1","recIDFreq":"2"}

Reco: 2010 Honda CR-V LX SUV, Crossover

Score: 0.038

Category: used cars & trucks

Location: Ottawa

Posted: 5 months ago (2015-11-25T02:19:37+0000)

State: ACTIVE

Description: Excellent Reliable Vehicle, 2010 HONDA CRV-LX- black with black interior-164km, $11000 Firm---cheapest price in ontario - 4 CYLINDER AUTOMATIC, PO...

Recommendation metadata: {"adIdFreq":"8","coVisit":"4","recIDFreq":"13"}

Advice to the masses

- Will covisitation work for you?
  - This depends on your domain and one should start by analyzing what ‘co’ metric you are going to use. i.e. views? Likes? Purchases? Etc…
  - If collaborative behaviour is small for that metric then don’t bother going any further. OR increase your timeframe.
  - Also works for traditional ‘users’. You don’t need it ‘session’ based.
- Covisitation may not be enough on its own
  - A good example of another domain using covisitation a key signal for recommendations is FOURSQUARE (http://engineering.foursquare.com)

Thank you

Collaborative Recommendations

ECG Finding Science (FiSci) Team

June 29, 2016

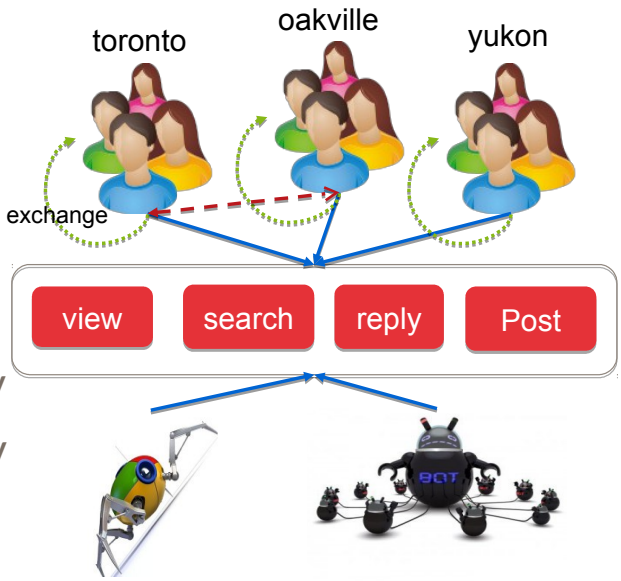
Kijiji Marketing initiative / personalization

Format

- Introductions
- Framing the problem and thinking through it.
- Short break
- Taking the solution to spark (the code)
- Other things involved in the overall system

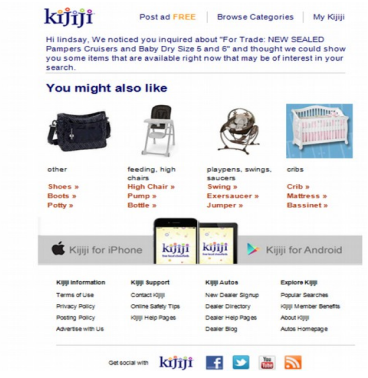
What all our markets have in common…

- Local Community driven
- Highly anonymous interactions / low touch
- Diverse users with ever changing/different needs
- Locality sensitive inventory
- Rapidly changing inventory
- Non-human traffic

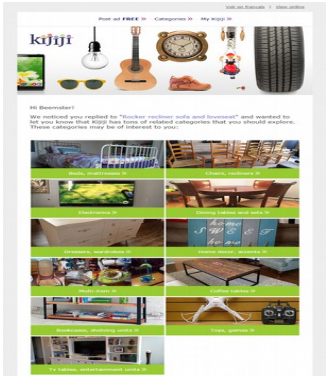


Common Goal

- Enrich the current *offsite re-engagement* experience, driving users back to our platform to *discover* new content and/or help them *succeed*.



Live item level similar / related per category



Idea

- Recommend items for re-engagement to buyers that *reply* to an item within a given threshold. Use *reply* as the *trigger* signal. Recommendations based off the item replied to.
  - Measure the overall open rate and success of the version of those emails against the baseline
  - Measure the session activity of those users arriving from that email against the baseline and measure the conversion rate on specific items clicked on from the email.
  - Triggered on personalized action, but can we utilize collaborative activity of other users on the site/app?

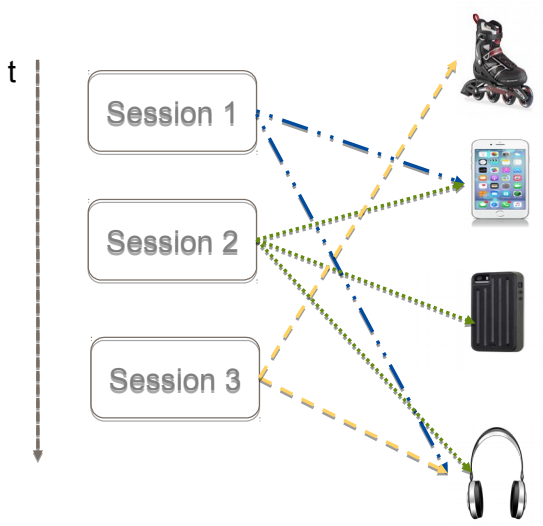
FiSci recommendations

- First pass based on a normalized *co-visitation* algorithm to capture the *collaborative* aspect
- Aiming to capture more '*similarity*' than '*related' but capable of* tuning per category and category trees and other dimensions
- Constraints in place
- Be able to generate recommendations at scale multiple times, quickly and reliably.

Recommendation Techniques

- User / Item Based (Collaborative Filtering)
  - Predictive *model / ML / ALS Scaling can be tough and data very sparse. Doesn't work if you have no underlying STRONG linear latent factors. i.e no strong primary source(s) of variance.*
- Collaborative in general
  - Recommendations determined by *actions* of *entities* within the domain
- Content / Rule Based
  - Intersecting item feature space with user feature space
  - Merchandising / Decision Trees
- Who cares based ☺
  - Almost all end up being hyrbids anyway because most don't perform / can't be tuned optimally on their own

Covisitation



Co-pairs

Recos

Covisitation Memory-Based Algorithm For Recommendation

This memory-based algorithm is simply defined as:-

For a given time period (usually 24 to 48 hours) we count for each pair of adIds (t_i, t_j) in one session how often they were co-visited across all other sessions. Denoting this co-visitation count by c_{ij}, we define the related score of item t_j to base item t_i as:

$$r(t_i, t_j) = \frac{c_{ij}}{f(v_i, v_j)}$$

where c_i and c_j are the total occurrence counts across all sessions for item t_i and t_j, respectively. $f(v_i, v_j)$ is a normalized function that takes the "global popularity" of both the seed item and the candidate item into account.

[1] Davidson, James, et al. "The YouTube video recommendation system. "*Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010.
[2] Carbone, Paris, and Vladimir Vlassov. "Auto-Scoring of Personalised News in the Real-Time Web: Challenges, Overview and Evaluation of the State-of-the-Art Solutions." *Cloud and Autonomic Computing (ICCAC), 2015 International Conference on*. IEEE, 2015.
[3] Das, Abhinandan S., et al. "Google news personalization: scalable online collaborative filtering." *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007.
[4] Bates, K.M., Paas, J., Wang, B., Xu., B. and Yousefi, P., Recommender System for on-line articles and documents, *Google Patents (http://www.google.com/patents/US20090300547)*, 2009, (US Patent App. 12/360, 516).

(This approach has already proved useful for Google News aggregation and for Youtube Collaborative Video Recommendation)
[5] Malik, Z. K., & Fyfe, C. (2012). Review of Web Personalization. *Journal of Emerging Technologies in Web Intelligence*, *4*(3), 285-296.
[6] Malik, Z. K., & Connolly, T. M. (2012). A new personalized approach in affiliate marketing. *e-society*, 235.
[7] Malik, Zeeshan Khawar, Colin Fyfe, and Malcolm Crowe. "Priority recommendation system in an affiliate network." *Journal of Emerging Technologies in Web Intelligence* 5.3 (2013): 222-229.

Computed Symmetrical Covisitation Matrix

ADID / ADID	A_1	A_2	A_3	A_4	A_5	.	.	.	.	A_N
A_1	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	.	.	.	.	C_{1N}
A_2	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	.	.	.	.	C_{2N}
A_3	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}	.	.	.	.	C_{3N}
A_4	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}	.	.	.	.	C_{4N}
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
A_N	C_{N1}	C_{N2}	C_{N3}	C_{N4}	C_{N5}	.	.	.	.	C_{NN}

Primary Reasons behind Selecting Covisitation Memory Based Algorithm

1. Simple and quickest
2. Scalable
3. Contains Efficient Collaborative Personalization Abilities
4. Can be used as a Ranker (for Ad Items) for various other computational purposes.
5. Can be easily integrated with other scalable and efficient Model Based Approaches
6. Maintain Strong Similarity Between Ad Items

Step one – Computing Covisits

Step 2 – Computing Covisits

Session View Pairs

session_id	test_id	rec_id
s1	1	2
s1	2	1
s2	1	2
s2	2	1
s4	3	2
s4	2	3

```
select test_id, rec_id, count(1)
from view_pairs
groupby test_id, rec_id
```

Covisit Triples

test_id	rec_id	covisits
1	2	2
2	1	2
2	3	1
3	2	1

Recos = group by test_id

1 → { 2 }

2 → { 1, 3 }

3 → { 2 }

Step 3 – Can we do better?

- **Symmetric** Relation
  - (a,c) ⇔ (c,a)
  - exploiting symmetry can HALVE your datasize.
  - exploiting symmetry can halve the aggregated key space
  - BIG win with BIG data
- **NOT Transitive**
  - (a,c) & (c,b) !-> (a,b)

- Work with this instead

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1q} \\ a_{21} & a_{22} & & a_{2q} \\ \vdots & & \ddots & \vdots \\ a_{q1} & a_{q2} & \cdots & a_{qq} \end{bmatrix}$$

Preview

- We will leave this out for time / simplicity, but the idea…*(given item id's with a strict partial ordering)*

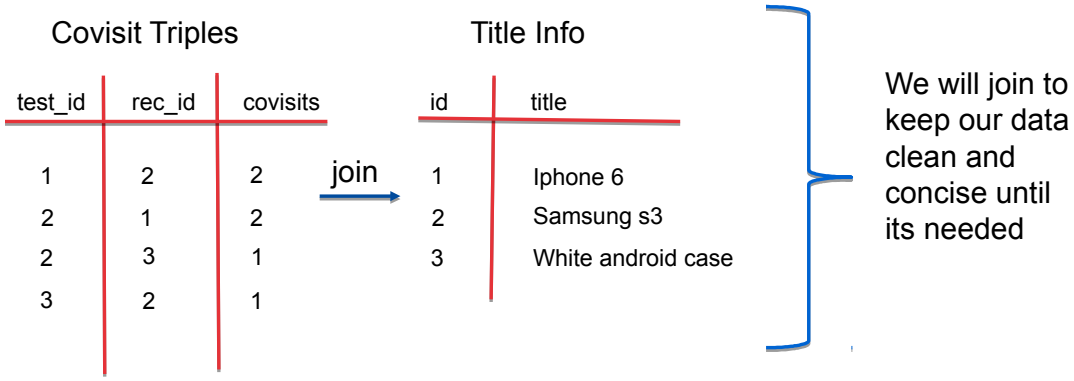
Will require double reduction in the final step. But still more efficient

What about ***f(v_i, v_j)*** ??

- Right now our recommendations for a given item are just an unordered set of covisited items
- Probably not enough given whatever your objective / requirements are.

- E.g
  - *f(v_i, v_j) = Cosine similarity of title v_i with title_vj*
  - *Assumes you carried the title along as a column of each item. problem?*
  - *Score of a given ad is then c_ij * f(v_i,v_j)*
  - *What would this do?*

Bringing in Score



Bringing in Score

Covisits w/ title

test_id	test_t	rec_t	rec_id	covisits
1	t1	t2	2	2
2	t2	t1	1	2
2	t3	t2	3	1
3	t3	t3	2	1

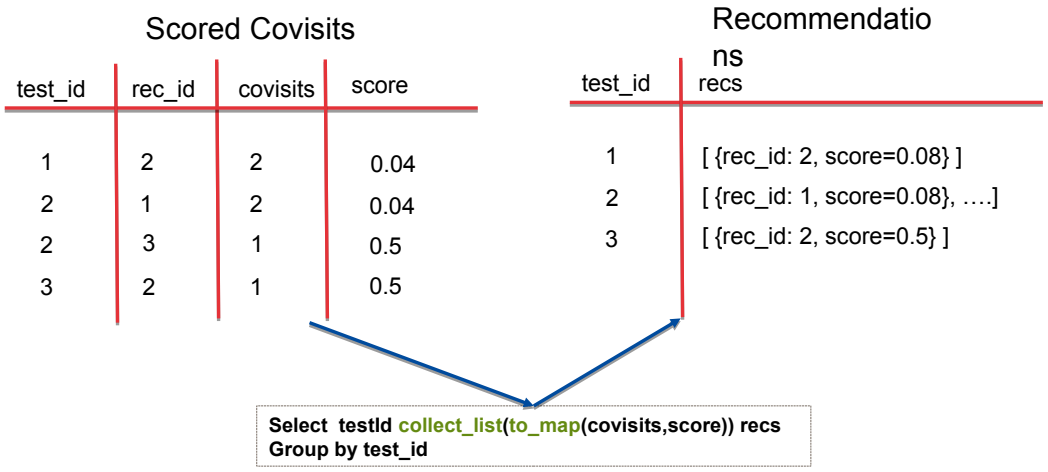
Select test_id, rec_id, covisits * coss(test_t,rec_t) score

Scored Covisits

test_id	rec_id	covisits	score
1	2	2	0.08
2	1	2	0.08
2	3	1	0.50
3	2	1	0.50

Just an example. But what's wrong here?

Final Table of Recommendations



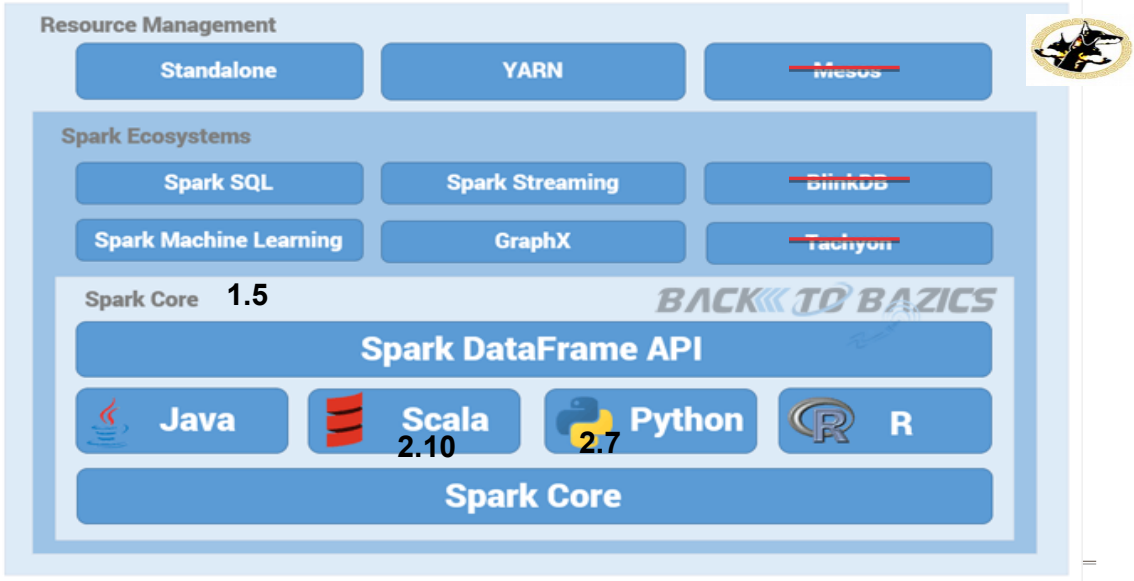
Questions and Break

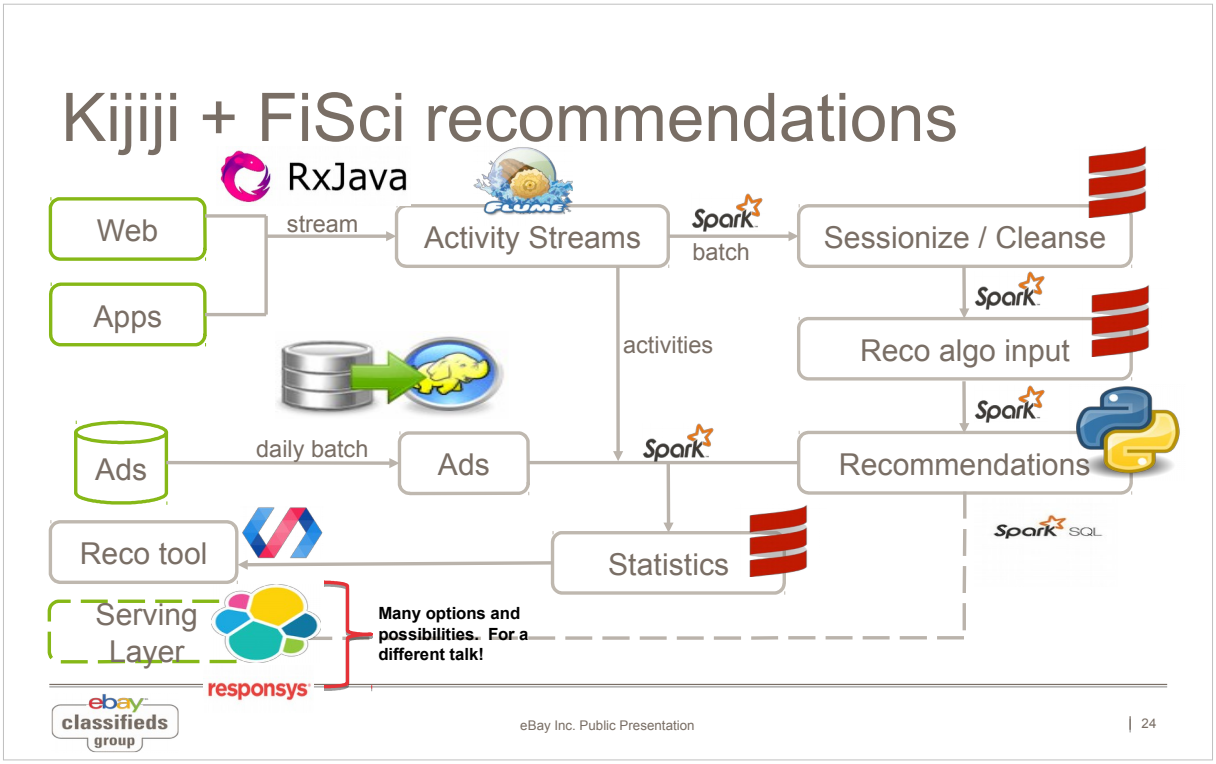
Taking Covisitation to Spark

Where the tires meet the road

Our Hadoop World

cloudera®





Right now all stats are reply-based, not ad-based, because of re-engagement email use case

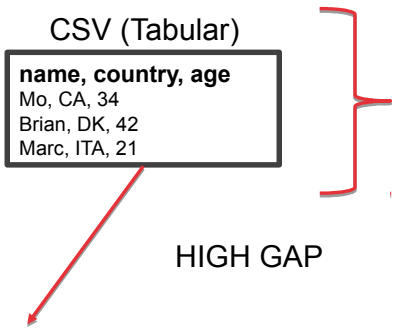
Where did python come from?

- The first project for the lead developer and Python was the most familiar language.
  - Quicker time to delivery
  - Didn't want to add extra complexity / risk to project
  - Easy to port later (We will most likely…)
  - Didn't see any red flags at the time. But of course we found some later ☺

Why SparkSQL / Dataframes?

- Lower Representational Gap
  - Less prone to error staying in our initial reasoning frame. DSL matches 1-1 (for most parts)
- Performance / Complexity
  - Dataframes (in many cases) can give you a major boost in performance for a few reasons…
  - In PySpark, we get less python interpretation and closer direct bytecode execution

Representational Gap (RDD)




```
people_rdd = sc.textFile(namesCsv).map(lambda line: line.split(","))
people_of_country = people_rdd.keyBy(lambda r: r[1]).mapValues(lambda v : v[0])
```

Representational Gap (Dataframe)

A couple lines of boiler plate in the application. But results in LOW gap for rest of the important area

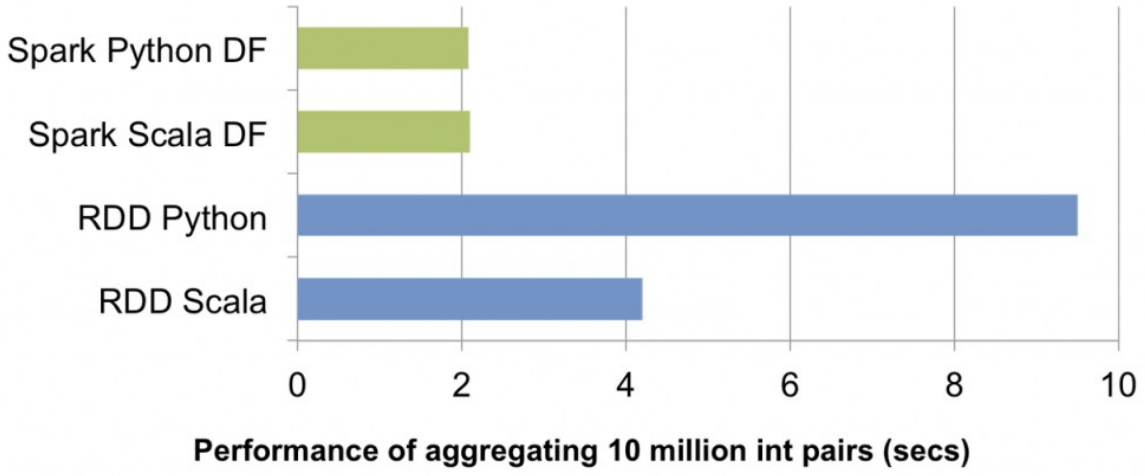
CSV (Tabular)

name, country, age
Mo, CA, 34
Brian, DK, 42
Marc, ITA, 21

```
sqlContext = SQLContext(sc)
people_df = df = sqlContext.read.format('com.databricks.spark.csv') \
      .options(header='true', inferschema='true') \
      .load('names.csv')

people_by_country_df = df.groupBy('country').agg(collect_list('name').alias('names'))

OR

people_df.registerTempTable("people")
people_df = sqlcontext.sql('SELECT country, collect_list(name) names FROM people GROUP BY name')
```

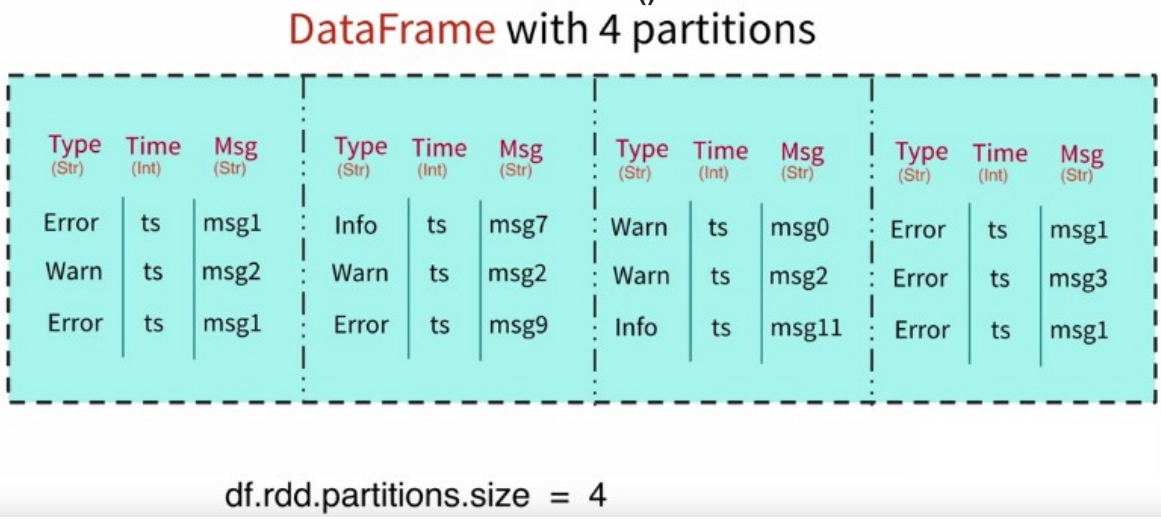
Performance

The infamous benchmark…not always the case. But usually



Performance

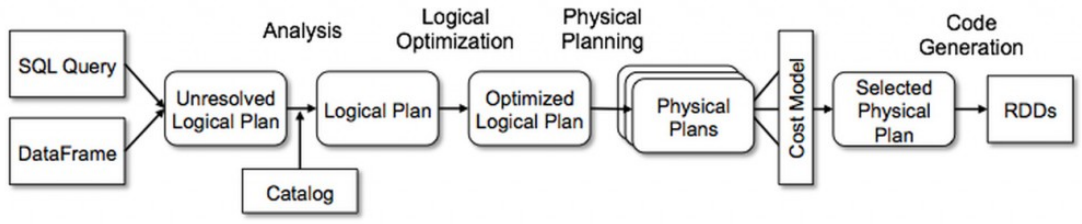
compress the column value based on the type defined in the column during cache()



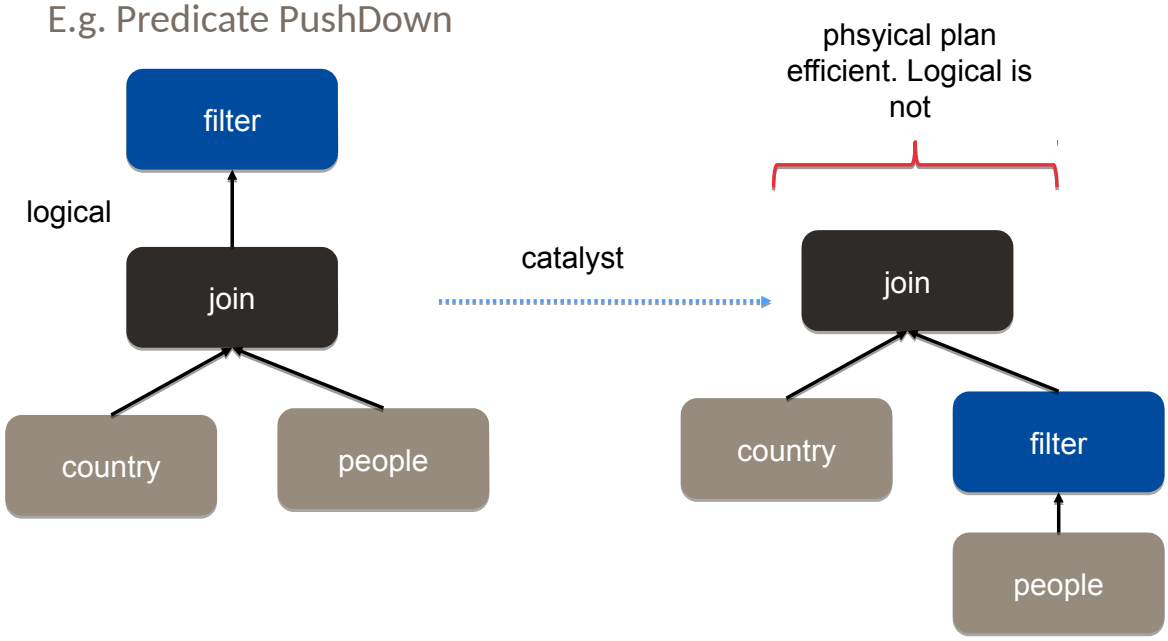
df.rdd.partitions.size = 4

Performance (The Catalyst Optimizer)

spark.sql.shuffle.partitions: 200 ? Huh?



E.g. Predicate PushDown



Starting at input

- Goal
  - A Flexible, compact format that can be easily extended and run through different algorithms / systems
  - Capture all view activities from the last 48 hours
  - Likely will be used for real time case if you want to extend with streaming
  - Difference in performance via Parquet? ***None***. Row orientied algorithm in our case. If projections of the input source are required however, then this would indeed be much less data read for nothing.

Starting At input

Magic of Cleansing / Sessionizing

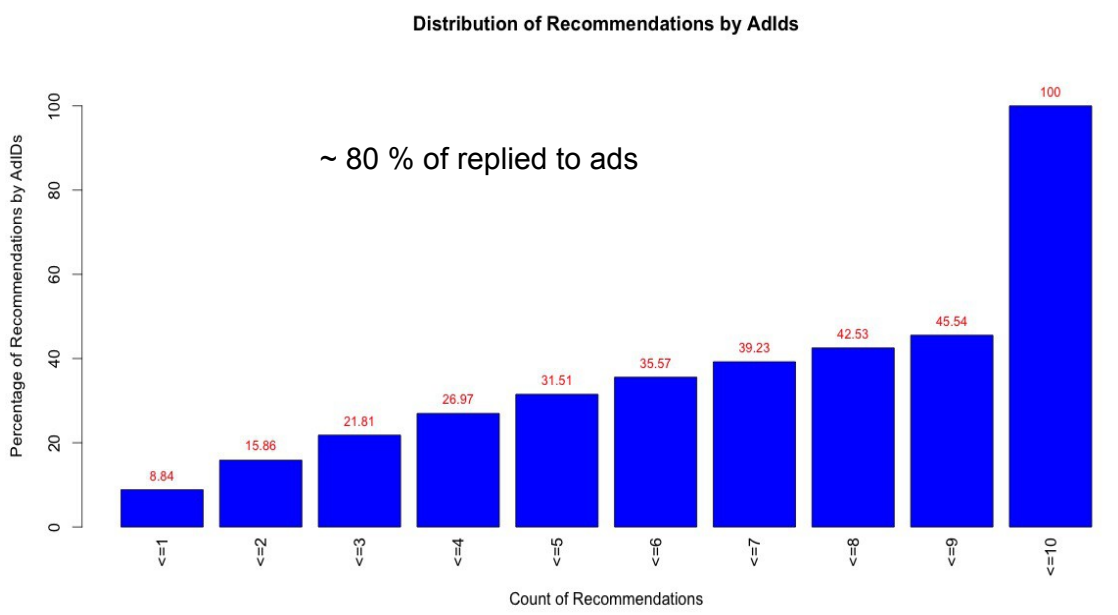
```
case class View(sessionId: String,
                adId: String,
                topAd: Boolean,
                hasImages: Boolean,
                l1Category: Int,
                leafCategory: Int,
                l1Location: Int,
                leafLocation: Int,
                userId: String,
                platform: String,
                creationDate: Long,
                lastEditDate: Long,
                viewTime: Long)
```

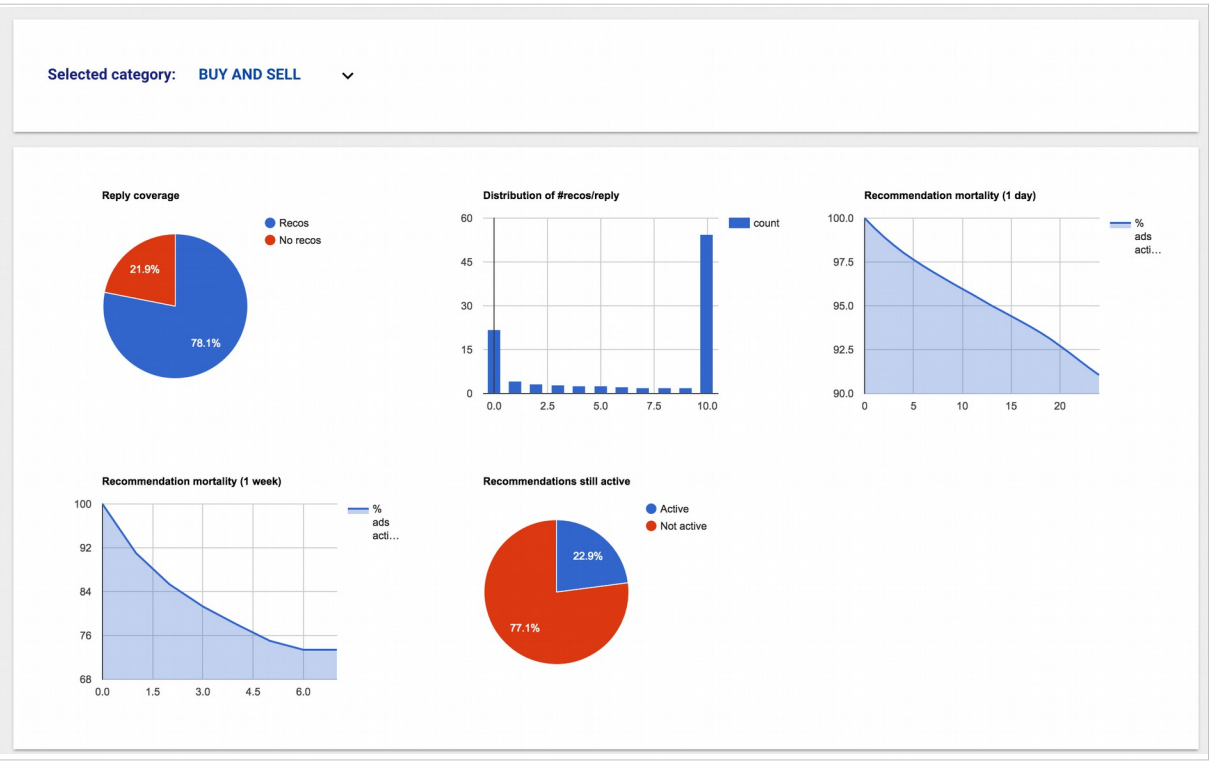
databricks™

```
import com.databricks.spark.avro._

val viewDF = viewRDD.toDF()

sqlContext.setConf("spark.sql.avro.compression.codec", "snappy")

viewDF.write.avro(outputPath)
```

Covisitation (The Code/Tests. Not a lot of it!)

Statistics / Judgements


Distribution of Recommendations by AdIds
~ 80 % of replied to ads
Percentage of Recommendations by AdIDs
0
20
40
60
80
100
8.84
15.86
21.81
26.97
31.51
35.57
39.23
42.53
45.54
100
<=1
<=2
<=3
<=4
<=5
<=6
<=7
<=8
<=9
<=10
Count of Recommendations



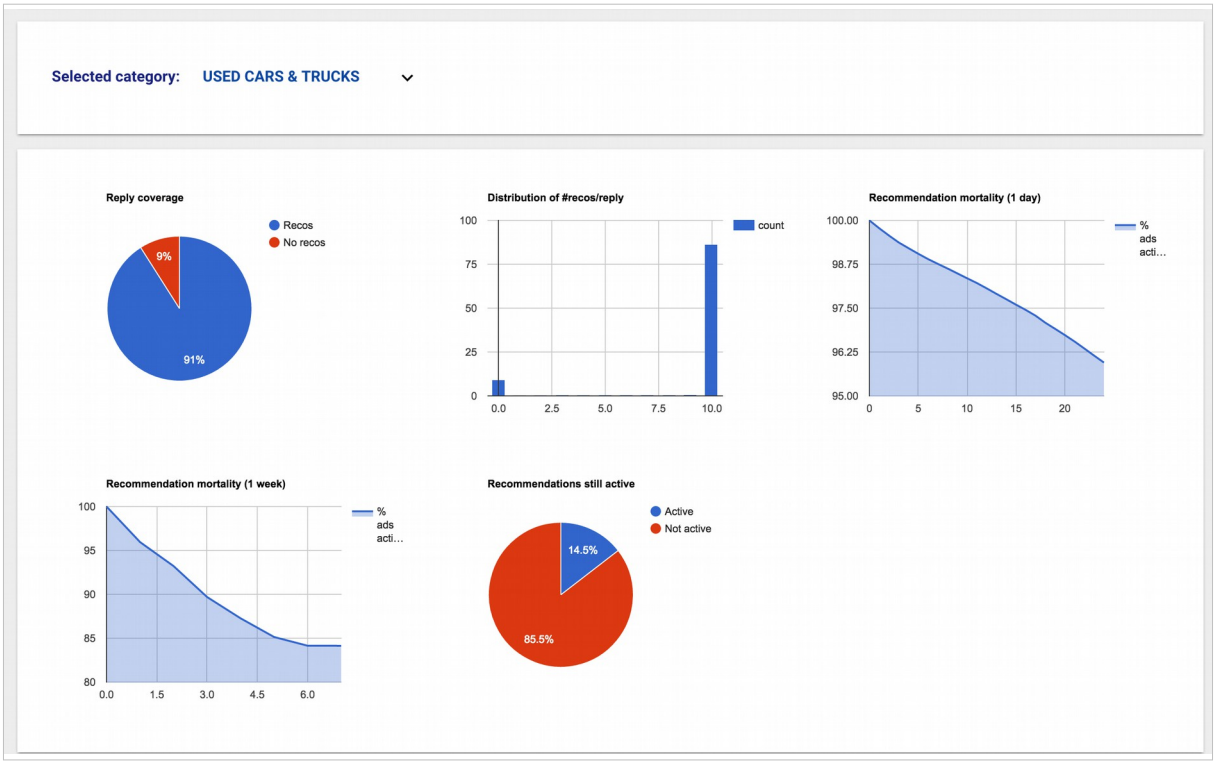
Algo sanity check.

Prototype, rough, put together last week.

Missing coverage due to ads without images being excluded

Divide by popularity to surface low exposure ads -> less mortality than overall kijiji ads

Caveat: mortality only includes known deletion dates. Only 6 days of data, not 7



Shows differences between L1s

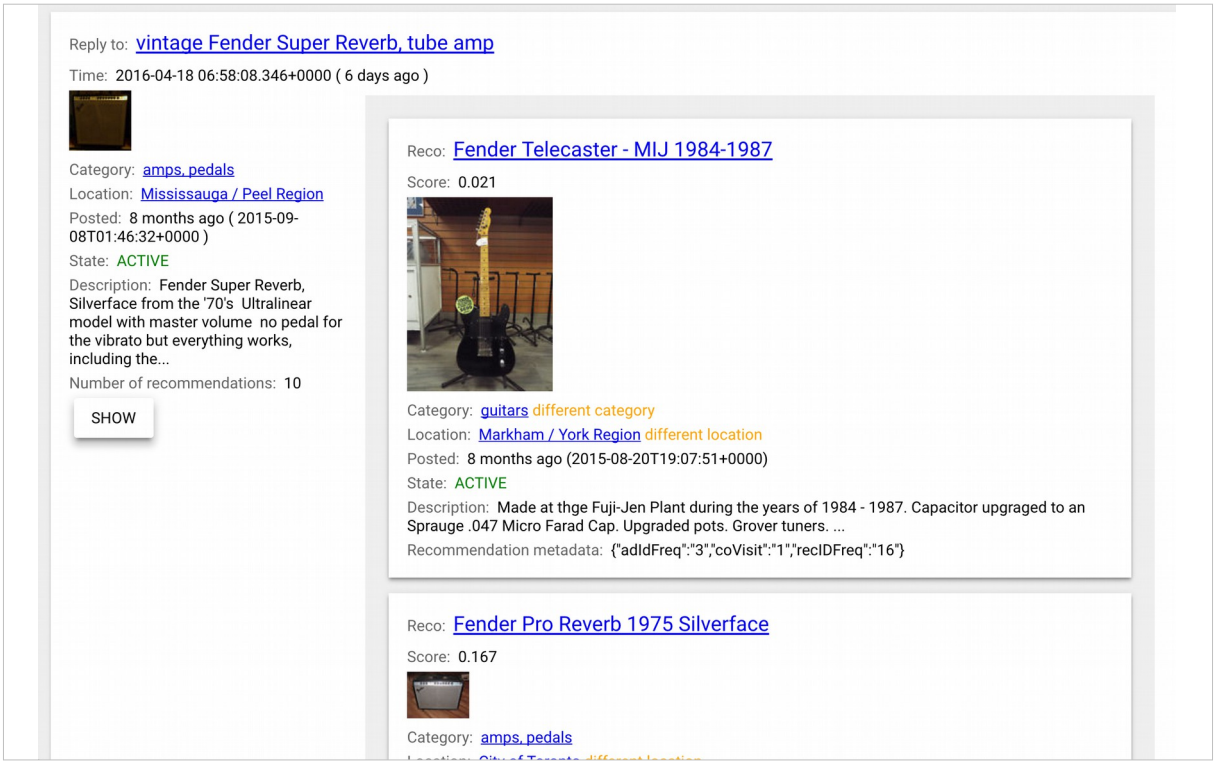
Gauging algorithm ‘quality’

Human judgment tool

Recos (ad id, [ad id]) -> no idea what’s going on

Qualitative human judgment

For us as well as Brian (marketing)



Contains algo score and metadata

Category/location re-sessionization (needed after boosting low exposure ads)

Next: compare algos side by side

Reply to: 2010 Honda CR-V LX DVD

Time: 2016-04-18 14:13:31.433+0000 (6 days ago)

Category: used cars & trucks

Location: Ottawa

Posted: a year ago (2015-02-17T07:24:18+0000)

State: ACTIVE

Description: <div sty...

Number of recommendations: 10

SHOW

Reco: 2012 Honda CR-V LX 5 SPD at 2WD (2)

Score: 0.063

Category: used cars & trucks

Location: Ottawa

Posted: 23 days ago (2016-04-01T19:21:19+0000)

State: ACTIVE

Description: <div sty...

Recommendation metadata: {"adIdFreq":"8","coVisit":"1","recIDFreq":"2"}

Reco: 2010 Honda CR-V LX SUV, Crossover

Score: 0.038

Category: used cars & trucks

Location: Ottawa

Posted: 5 months ago (2015-11-25T02:19:37+0000)

State: ACTIVE

Description: Excellent Reliable Vehicle, 2010 HONDA CRV-LX- black with black interior-164km, $11000 Firm---cheapest price in ontario - 4 CYLINDER AUTOMATIC, PO...

Recommendation metadata: {"adIdFreq":"8","coVisit":"4","recIDFreq":"13"}

Advice to the masses

- Will covisitation work for you?
  - This depends on your domain and one should start by analyzing what ‘co’ metric you are going to use. i.e. views? Likes? Purchases? Etc…
  - If collaborative behaviour is small for that metric then don’t bother going any further. OR increase your timeframe.
  - Also works for traditional ‘users’. You don’t need it ‘session’ based.
- Covisitation may not be enough on its own
  - A good example of another domain using covisitation a key signal for recommendations is FOURSQUARE (http://engineering.foursquare.com)

Thank you