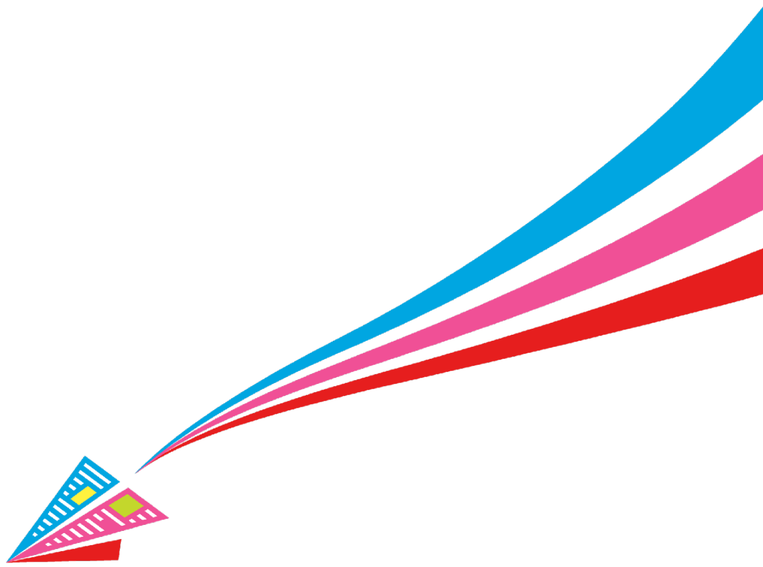


# Gunosyにおける AWS上での自然言語処理・機械学習の活用事例

株式会社Gunosy 開発本部データ分析部
大曽根 圭輔

Gunosy Inc.
2017.5

# 自己紹介

大曽根 圭輔 @dr_paradi 博士 (工学)

主な業務:
ニュースパス(明日でリリース1周年！)
のユーザ行動分析、記事配信アルゴリズム構築


大学時代の専攻:

人工知能の氷河期に
ファジィ理論の応用をやってました

# 株式会社Gunosy –「情報を世界中の人に最適に届ける」

Gunosyは 情報キュレーションサービス「グノシー」と
2016年6月1日にKDDI株式会社と共同でリリースした
無料ニュース配信アプリ「ニュースパス」を提供する会社です
「**情報を世界中の人に最適に届ける**」をビジョンに活動しています


情報キュレーションサービス **「グノシー」**

ネット上に存在するさまざまな情報を、
独自のアルゴリズムで収集、評価付けを行い
ユーザーに届けます。

無料ニュース配信アプリ **「ニュースパス」**

600媒体以上のニュースソースをベースに、
新たに開発した情報解析・配信技術を用いて自動的に
選定したニュースや情報をお客さまに届けます。

# データ分析部

サービスのコア部分である記事配信アルゴリズム改善、
ユーザ行動分析を担当。ブログを開設しています

Gunosy
データ分析ブログ

http://data.gunosy.io/

2017-02-02

## さくっとトレンド抽出: Pythonのstatsmodelsで時系列分析入門

Python　分析ノウハウ　時系列分析

久しぶりの投稿になってしまいましたが、ニュースパス(現在CM放映中!!)開発部の大曽根です。 作業中はGrover Washington Jr のWinelightを聴くと元気が出ます。参加ミュージシャンが素晴らしいですね。

1. なぜ時系列分析をするのか
2. 季節調整
3. 実演
4. おまけ: 時間別に見てみる
5. まとめ
6. 今後

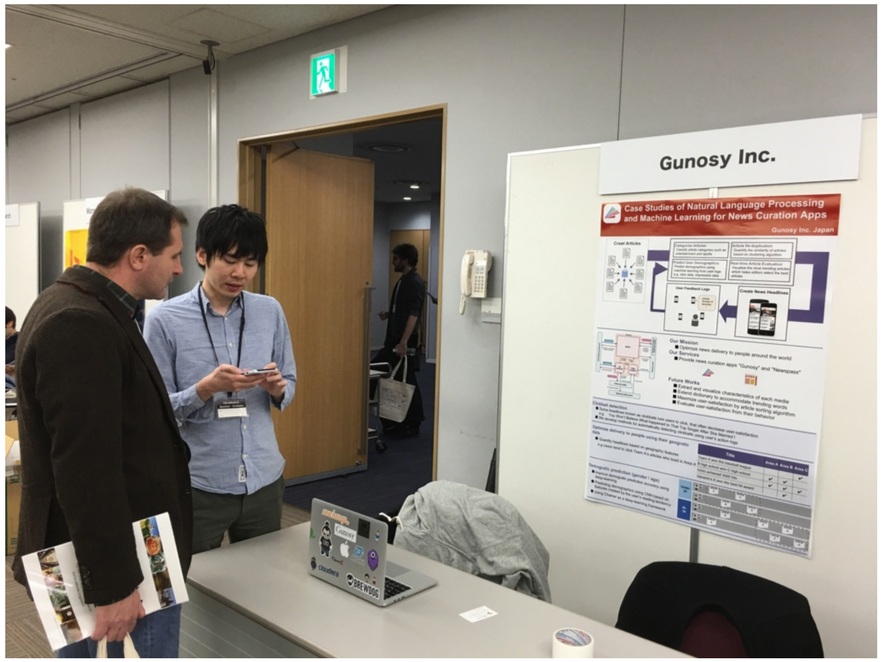
### なぜ時系列分析をするのか

数値を非常に重視している弊社では、数値を知るためのツールとしてRedashやChartioおよびSlackへの通知を活用しています。現在の数値を理解する上では、長期のトレンド(指標が下がっているのか、上がっているのか)を知ることが重要です。しかし、日々変化す

# データ分析部

サービスのコア部分である記事配信アルゴリズム改善、
ユーザ行動分析を担当。学会にも積極的に参加しています


JSAI 2017




COLING 2016

# はじめに

情報キュレーションサービスを対象に、AWS を用いて

人工知能技術を実サービスに応用する際の事例を紹介

機械学習ライブラリの充実により、ある程度の精度のもの(分類器などは)を作成するコストは下がっている

=> 実際のサービスで動かすことが重要

# はじめに

情報キュレーションサービスを対象に、AWS を用いて

人工知能技術を実サービスに応用する際の事例を紹介

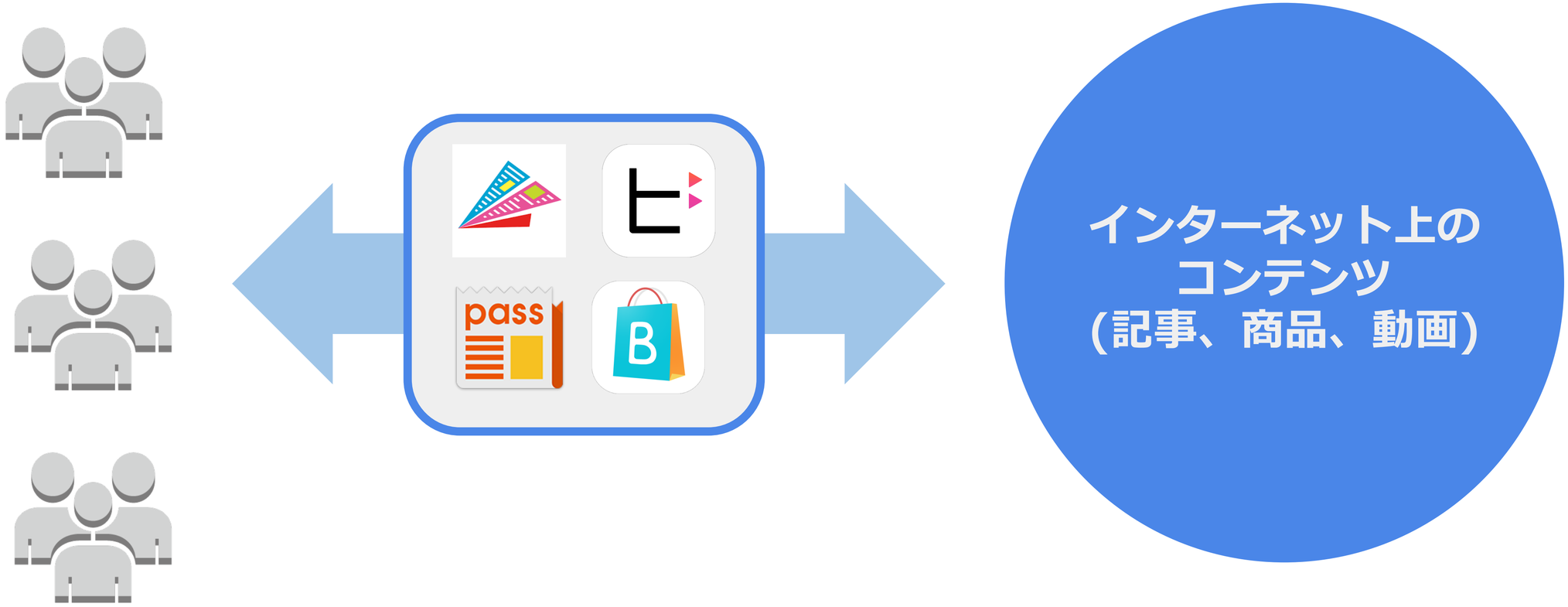
- 記事分類
- 属性推定 + スコアリング
- 効果測定 (ABテスト)

# 本日の発表の流れ

- Gunosyと機械学習
- 記事分類
- 属性推定 + スコアリング
- 効果測定 (ABテスト)

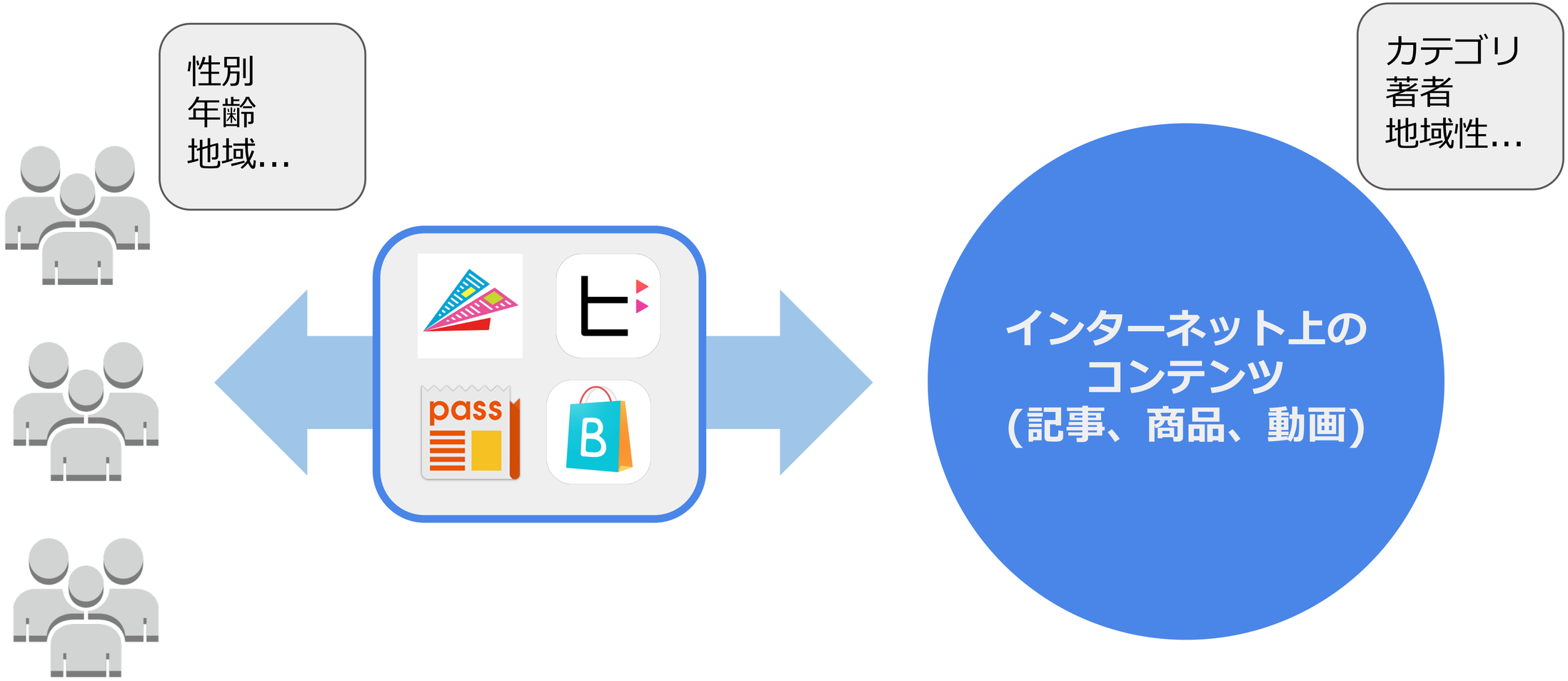
# Gunosyと機械学習

Gunosy では、ネット上に存在する様々な情報を独自のアルゴリズムで収集し、評価付けを行い、ユーザーに届けます



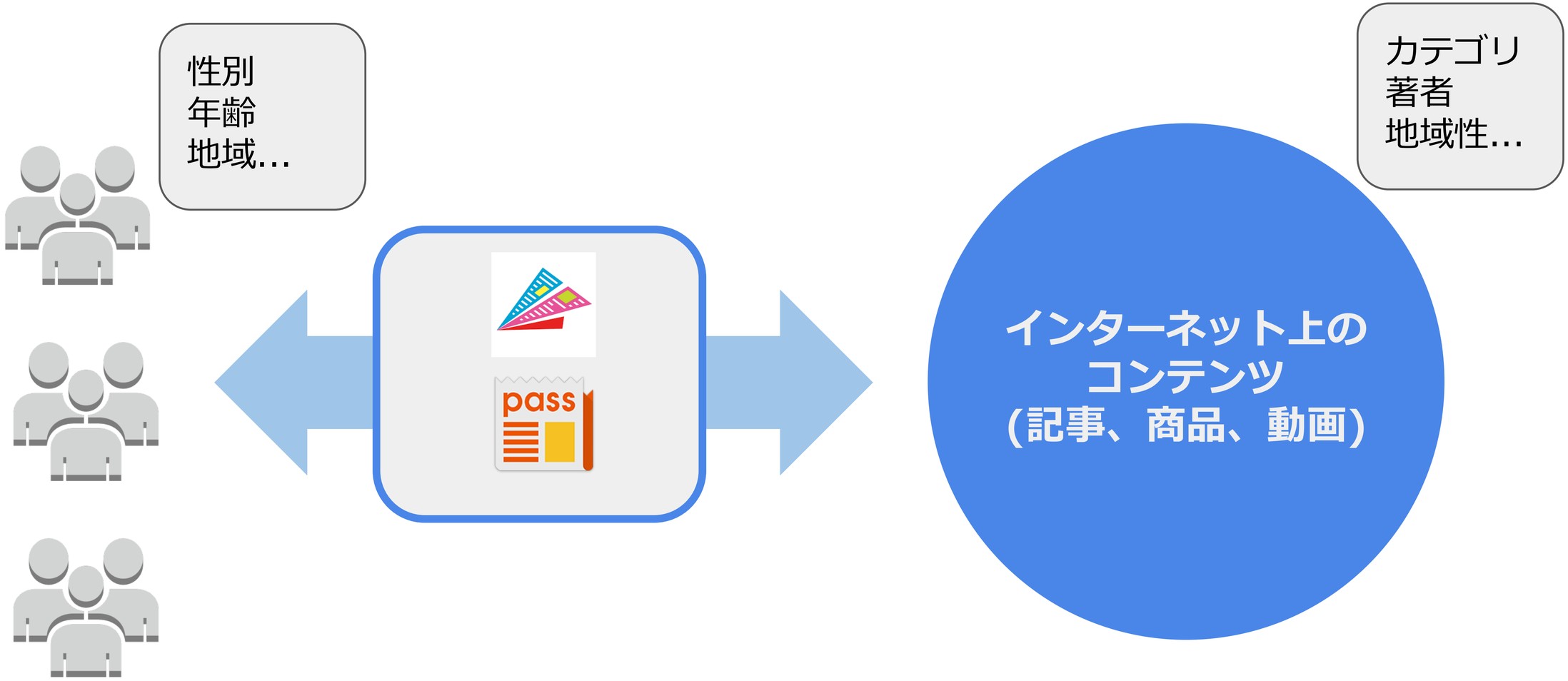
# Gunosyと機械学習

推定したユーザの属性、コンテンツの評価でマッチングの精度を向上

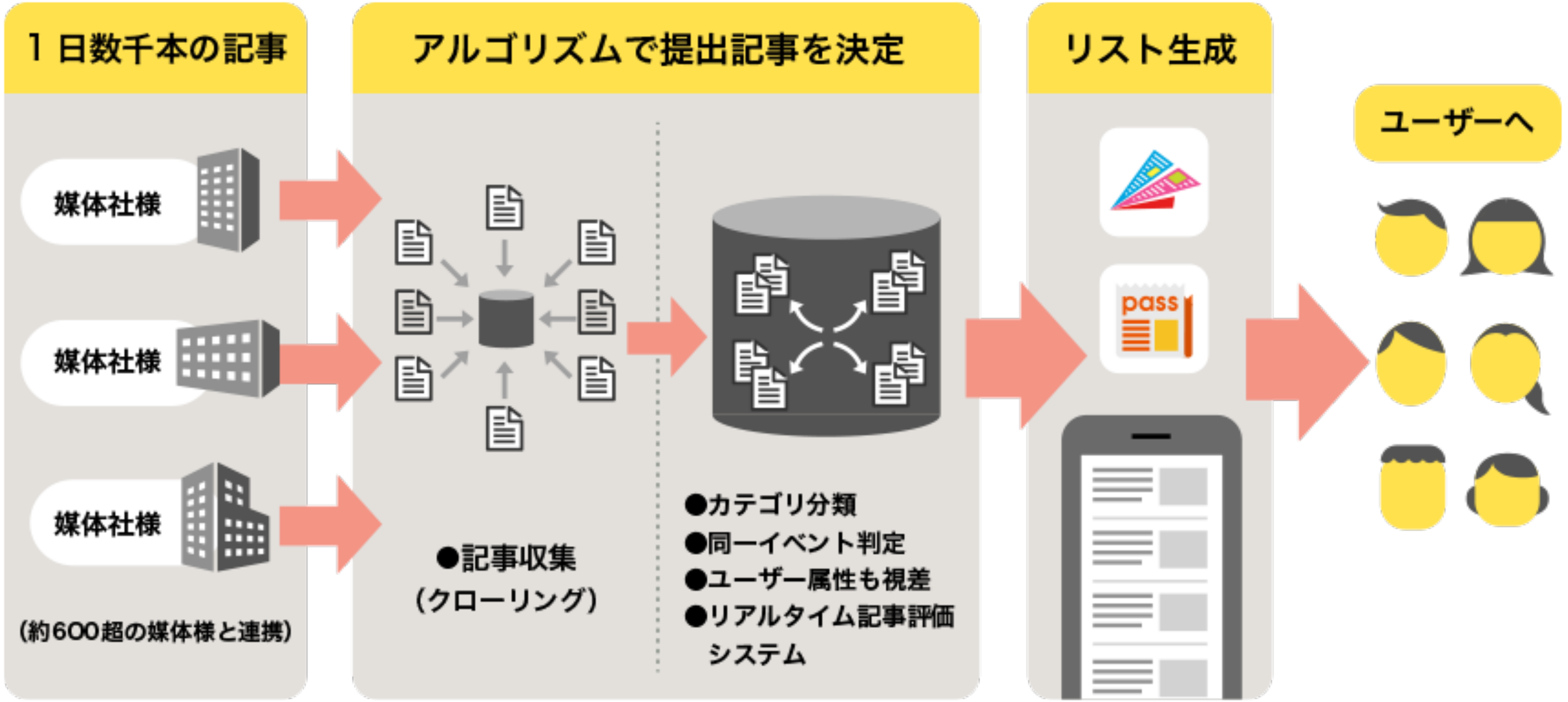


# Gunosyと機械学習

今回はニュース領域での活用事例を紹介



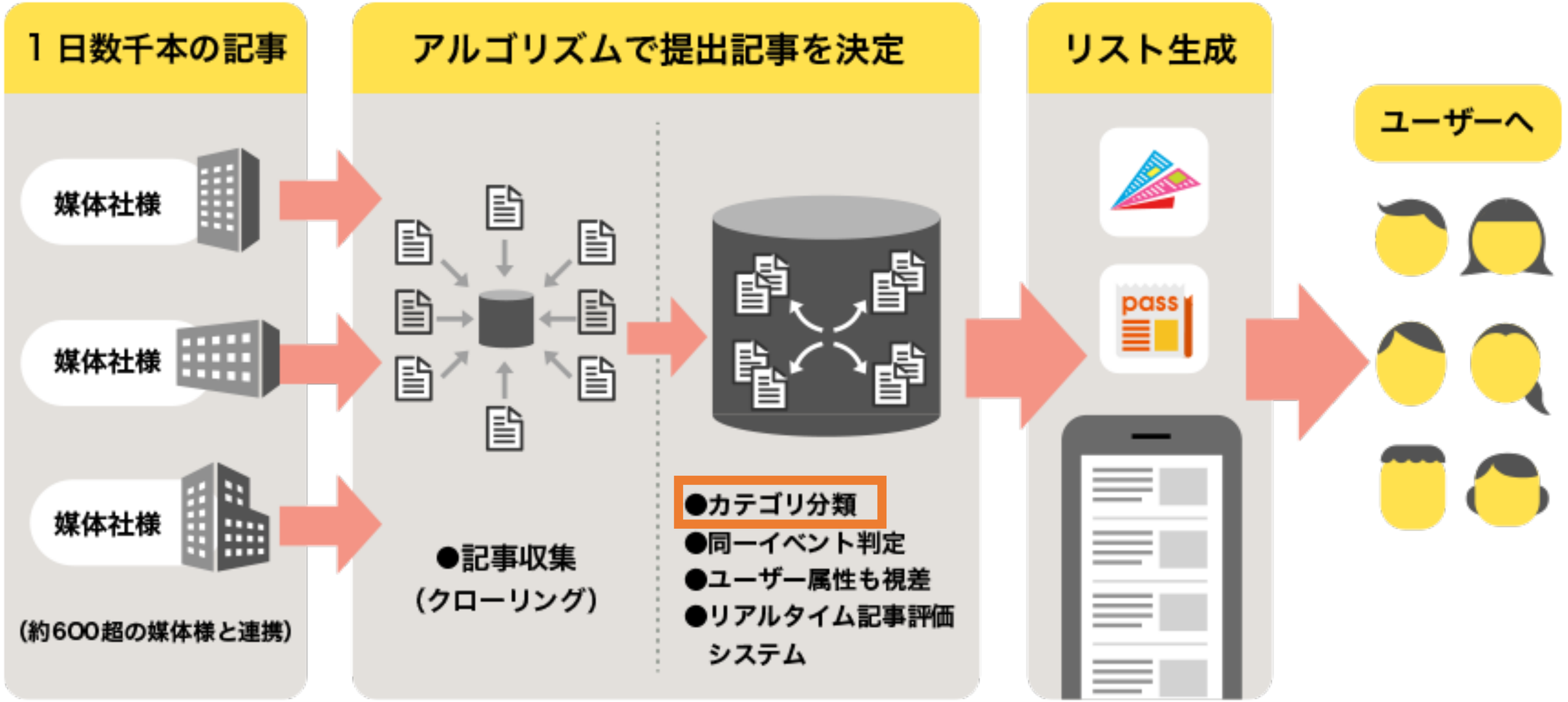
# 記事配信の流れ1



# 本日の発表の流れ

- Gunosyと機械学習
- 記事分類
- 属性推定 + スコアリング
- 効果測定 (ABテスト)

# 記事配信の流れ1

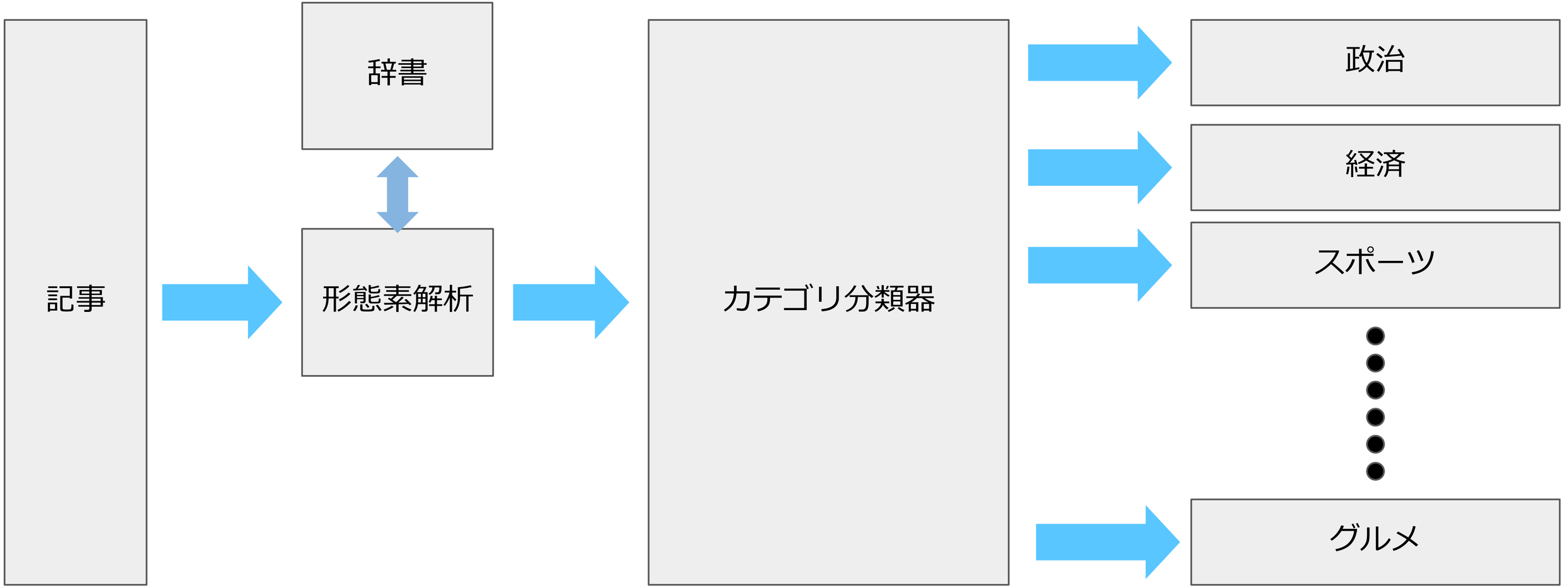


# 記事分類

それぞれの記事に対してどのカテゴリに分類されるかを判定
教師あり多クラス分類問題

| 記事の例 | 大カテゴリ | 中カテゴリ | 小カテゴリ |
|---|---|---|---|
| 日本代表のhogehogeが2試合ぶりゴール | スポーツ | サッカー | 日本代表 |
| fugafuga味のpiyopiyoが新発売!! | グルメ | 新商品 | |

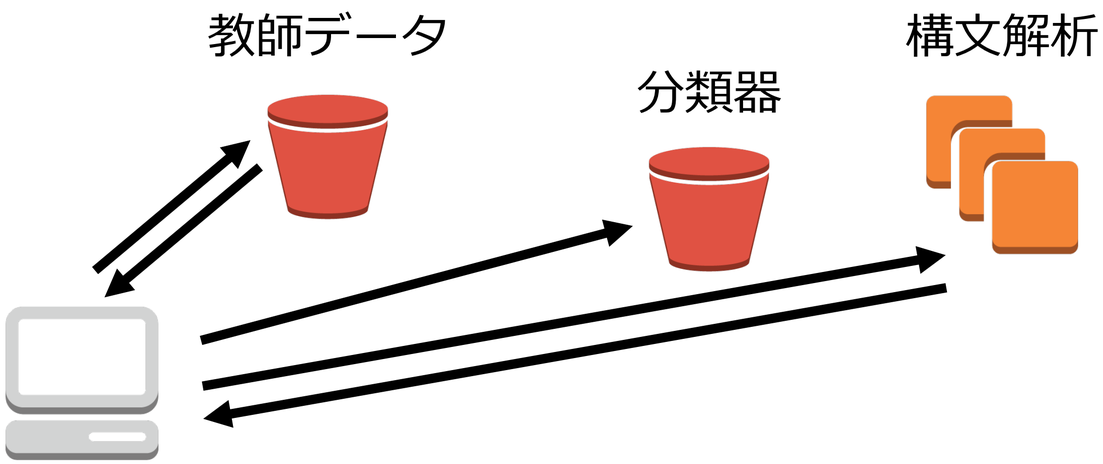
# 記事分類の流れ



# AWS上で動かしてみる

- クローラが記事を取得した際にカテゴリ分類APIにエンキューし分類、RDBに格納
  - 弊社のアプリは全てOpsWorksで管理
  - モデルはS3に保存しておき、deploy時に取得

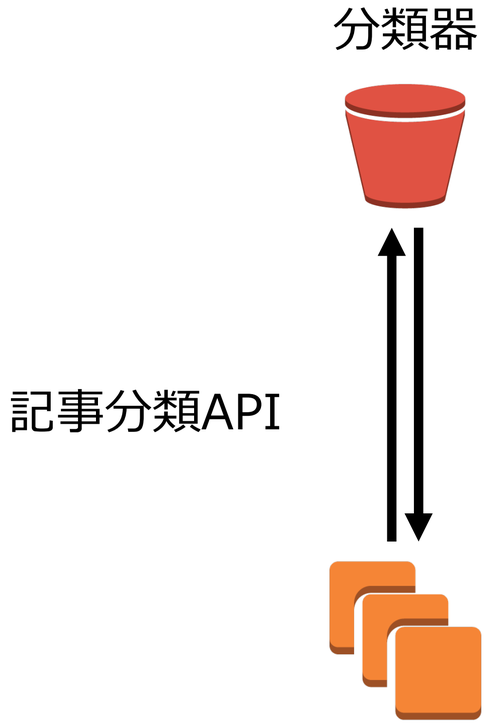
# AWS上で動かしてみる (学習時)



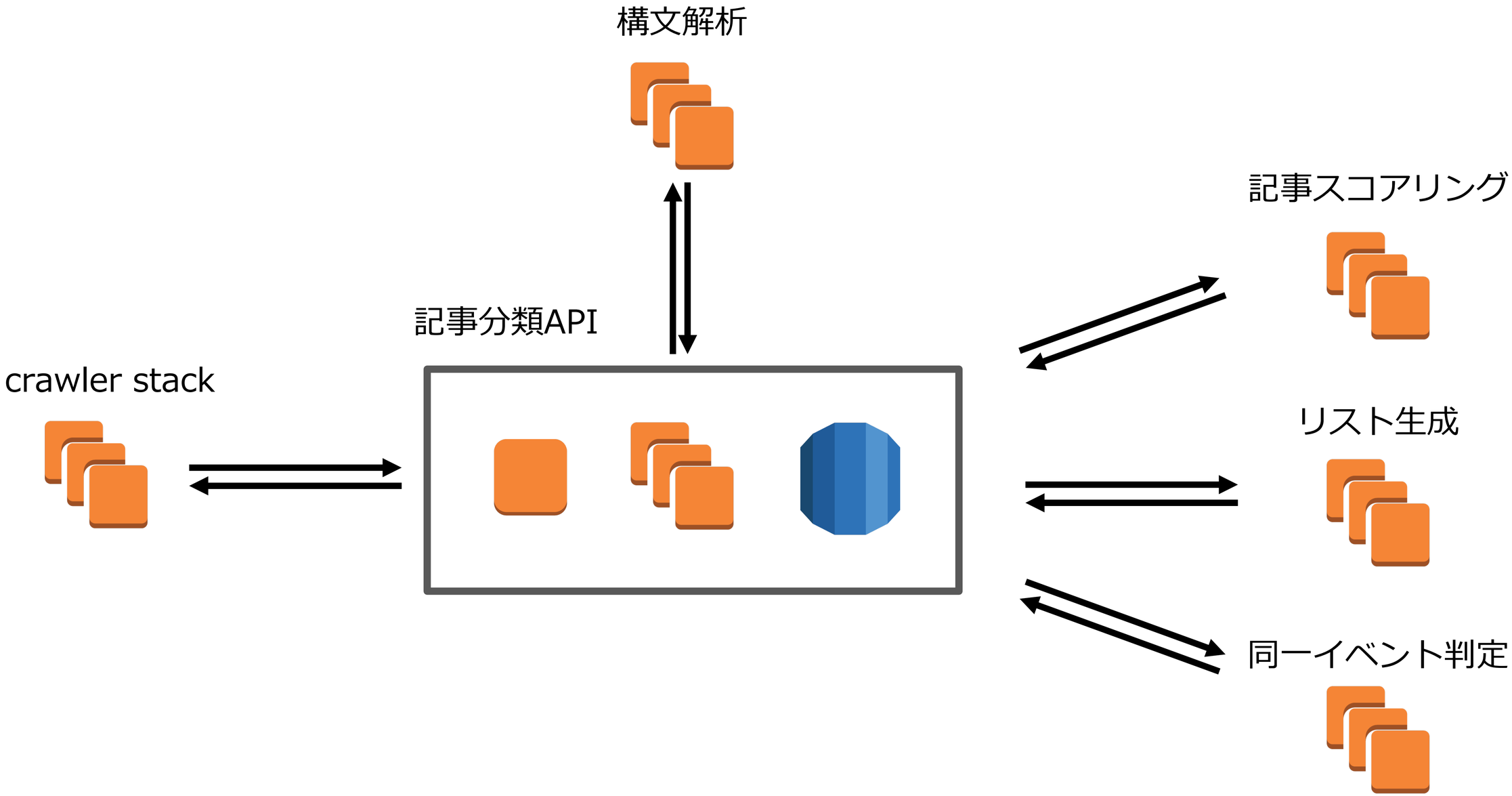

クラウドソーシングなどでカテゴリを付与された記事(教師データ)を元に、本文の構文解析を行い、に分類器を構築。S3にアップロード

# AWS上で動かしてみる (deploy時)

deploy時に各インスタンスが分類器を取得



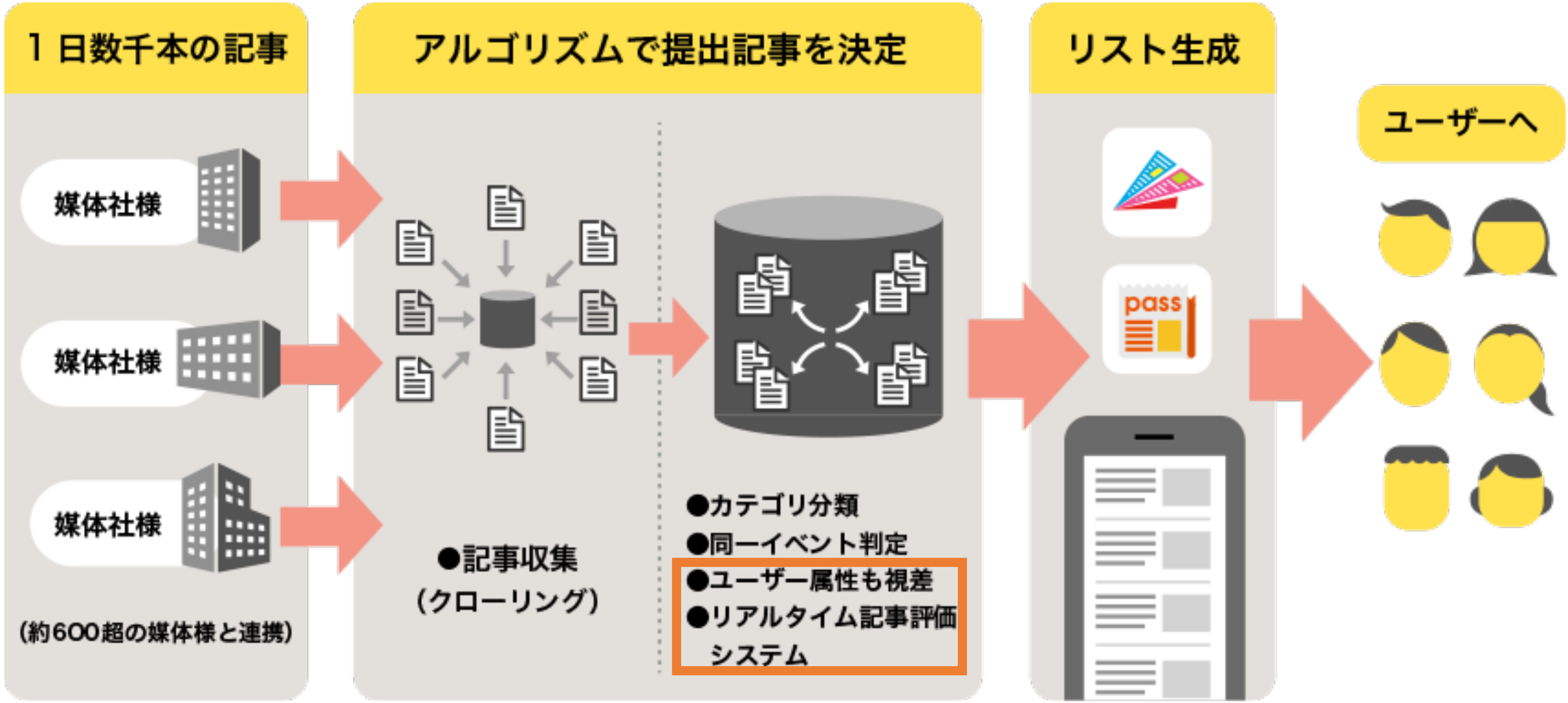
# AWS上で動かしてみる (deploy時)



# 本日の発表の流れ

- Gunosyと機械学習
- 記事分類
- 属性推定 + スコアリング
- 効果測定 (ABテスト)

# 記事配信の流れ1



# 属性毎の記事閲覧傾向の違い

- 性別
  - 男性はスポーツ記事をクリックしやすい
    - 野球は男性
    - フィギュアスケートは両方
  - 有名人の結婚・出産などのライフイベントは女性のほうがクリックしやすい
- 年齢
  - アーティストのニュースなどは年齢差が生まれやすい
- 地域
  - スポーツチームの勝敗や事件、イベントなどで地域毎でクリック傾向が異なる

# 属性毎の記事閲覧傾向の違い

1.直接ユーザにきく

- 入力ストレスなどでサービスから離脱してしまう恐れもある
- 全ユーザが入力してくれるわけではない
    - 上記を考慮して現時点では積極的には行わない

1.何かしらの手法で推定する

- ユーザが読んだ記事情報をもとに属性を推定

WebDB Forum 2016での発表より

http://data.gunosy.io/entry/webdbf2016

# 年齢推定の問題設定

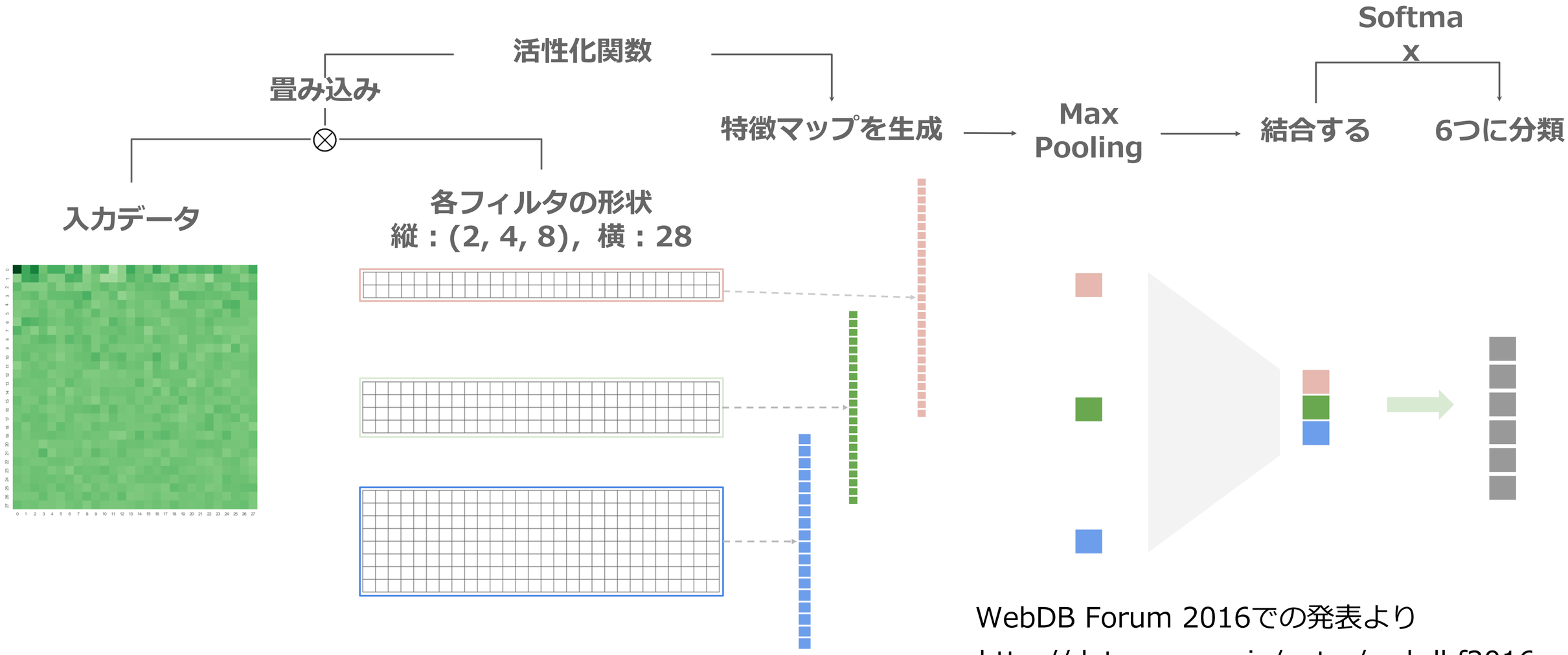
## クラス数

年齢を ~19歳，20~24歳，などに分けた多クラス分類問題

| 年齢の幅 | 値 |
|---|---|
| ~ 19 歳 | 0 |
| 20 ~ 24 歳 | 1 |
| 25 ~ 29 歳 | 2 |
| 30 ~ 39 歳 | 3 |
| 40 ~ 49 歳 | 4 |
| 50 ~ 歳 | 5 |

WebDB Forum 2016での発表より
http://data.gunosy.io/entry/webdbf2016

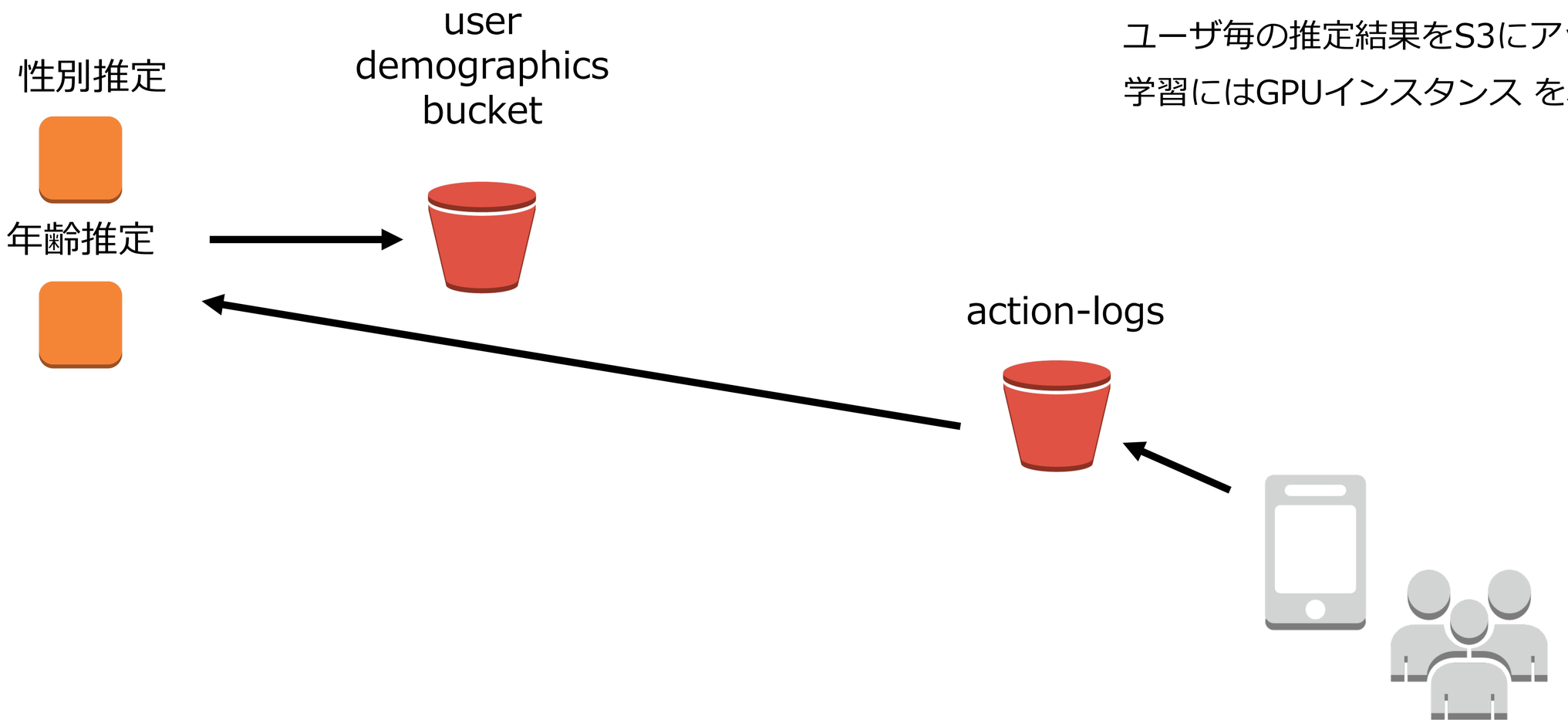
# 年齢推定にはCNN for NLP を応用




WebDB Forum 2016での発表より
http://data.gunosy.io/entry/webdbf2016

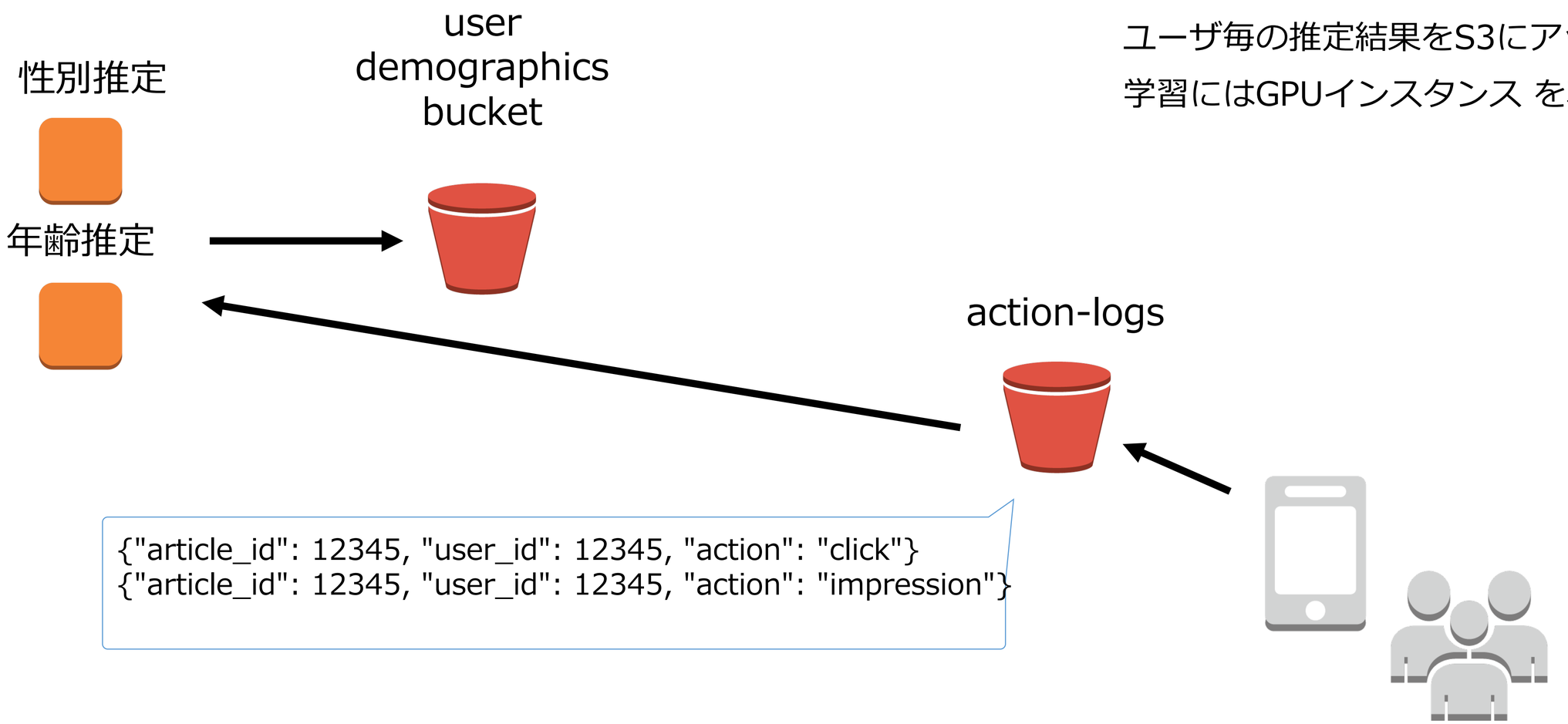
# AWS上で動かしてみる (推定)

ユーザの行動ログから推定モデルを構築し、
ユーザ毎の推定結果をS3にアップロード
学習にはGPUインスタンス を利用



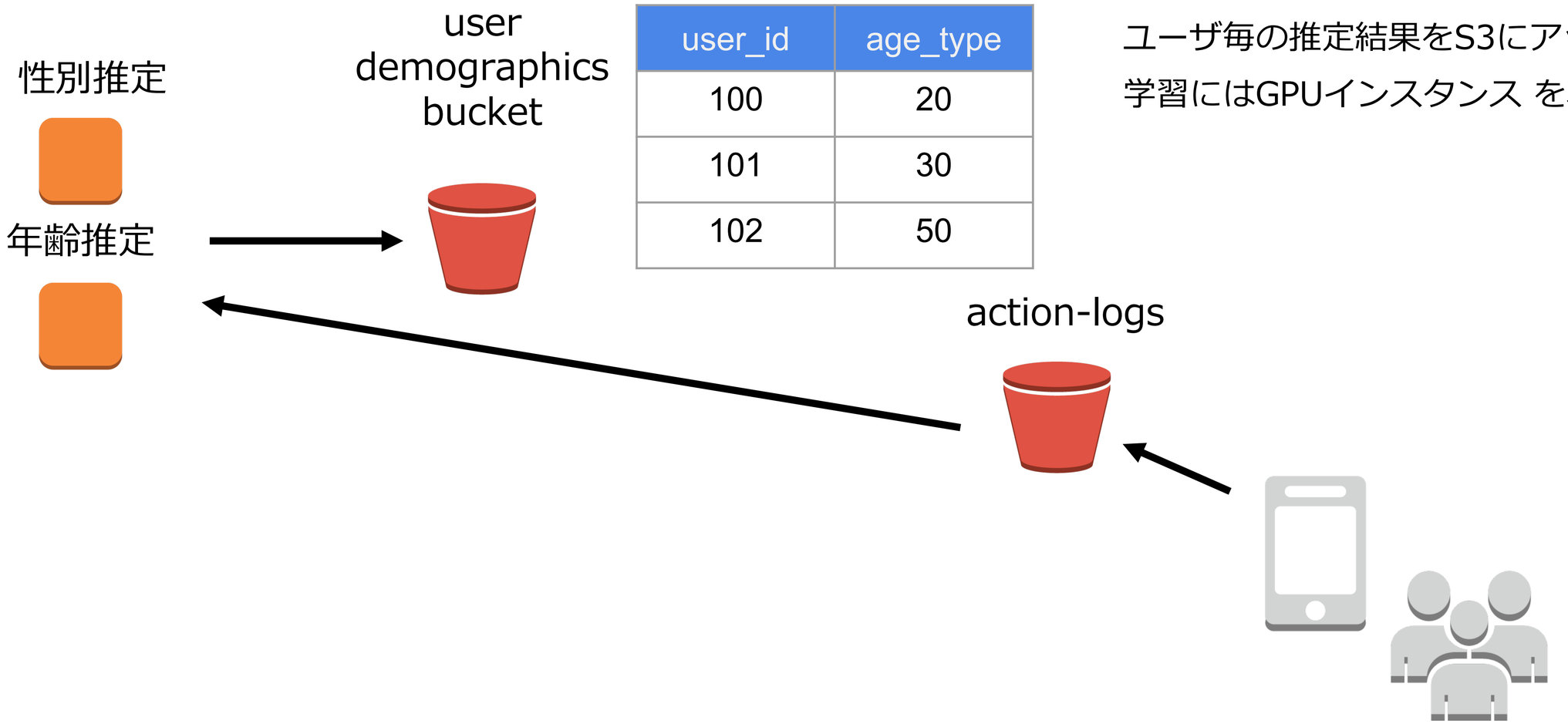
# AWS上で動かしてみる (推定)

ユーザの行動ログから推定モデルを構築し、
ユーザ毎の推定結果をS3にアップロード
学習にはGPUインスタンス を利用



# AWS上で動かしてみる (推定)

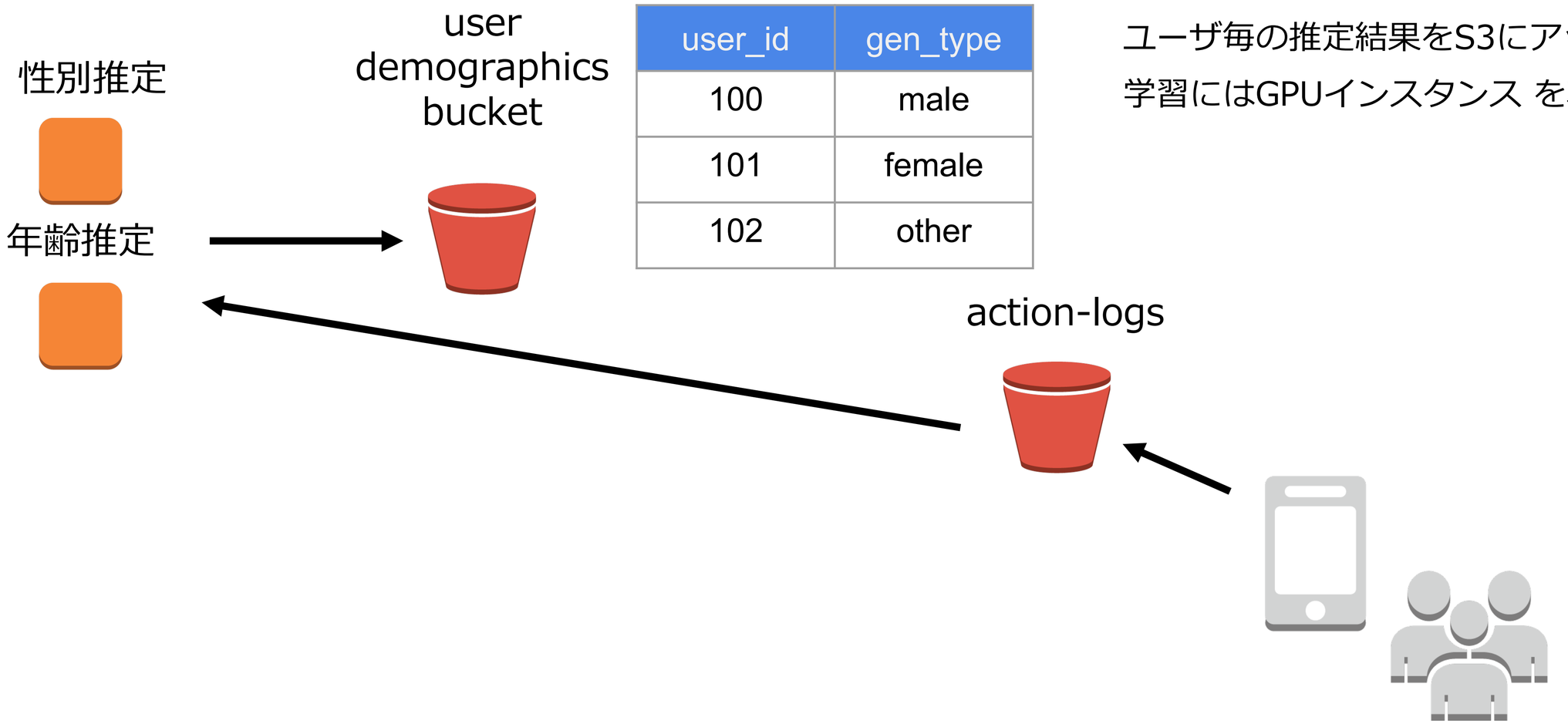
ユーザの行動ログから推定モデルを構築し、
ユーザ毎の推定結果をS3にアップロード
学習にはGPUインスタンス を利用




| user_id | age_type |
|---|---|
| 100 | 20 |
| 101 | 30 |
| 102 | 50 |

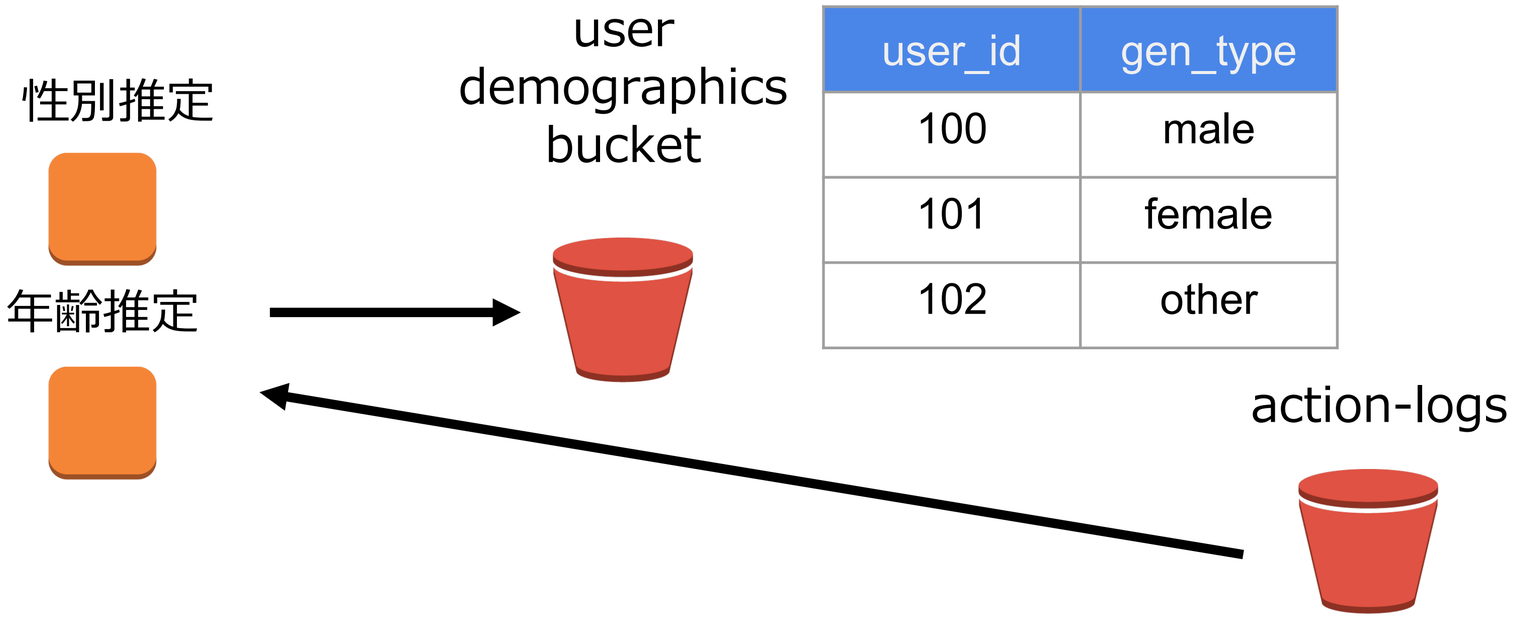
# AWS上で動かしてみる (推定)

ユーザの行動ログから推定モデルを構築し、
ユーザ毎の推定結果をS3にアップロード
学習にはGPUインスタンス を利用



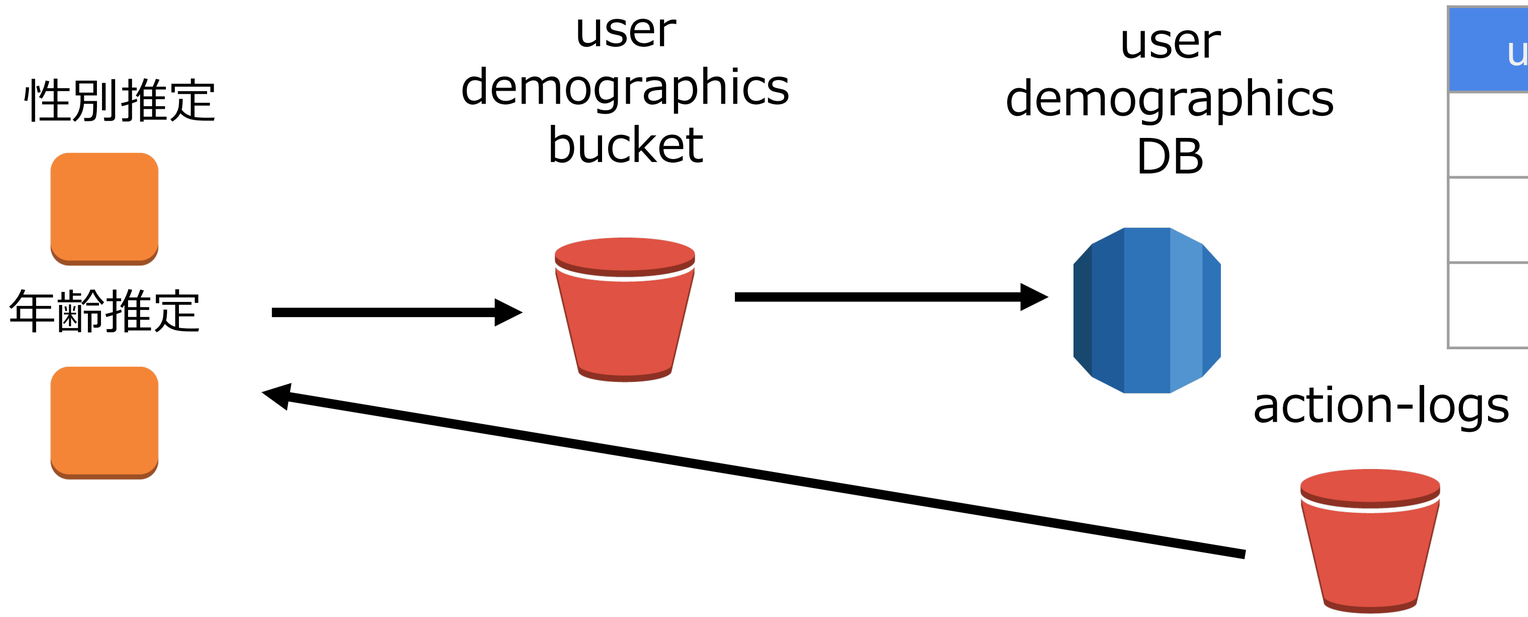

| user_id | gen_type |
|---|---|
| 100 | male |
| 101 | female |
| 102 | other |

# AWS上で動かしてみる (モニタリング)



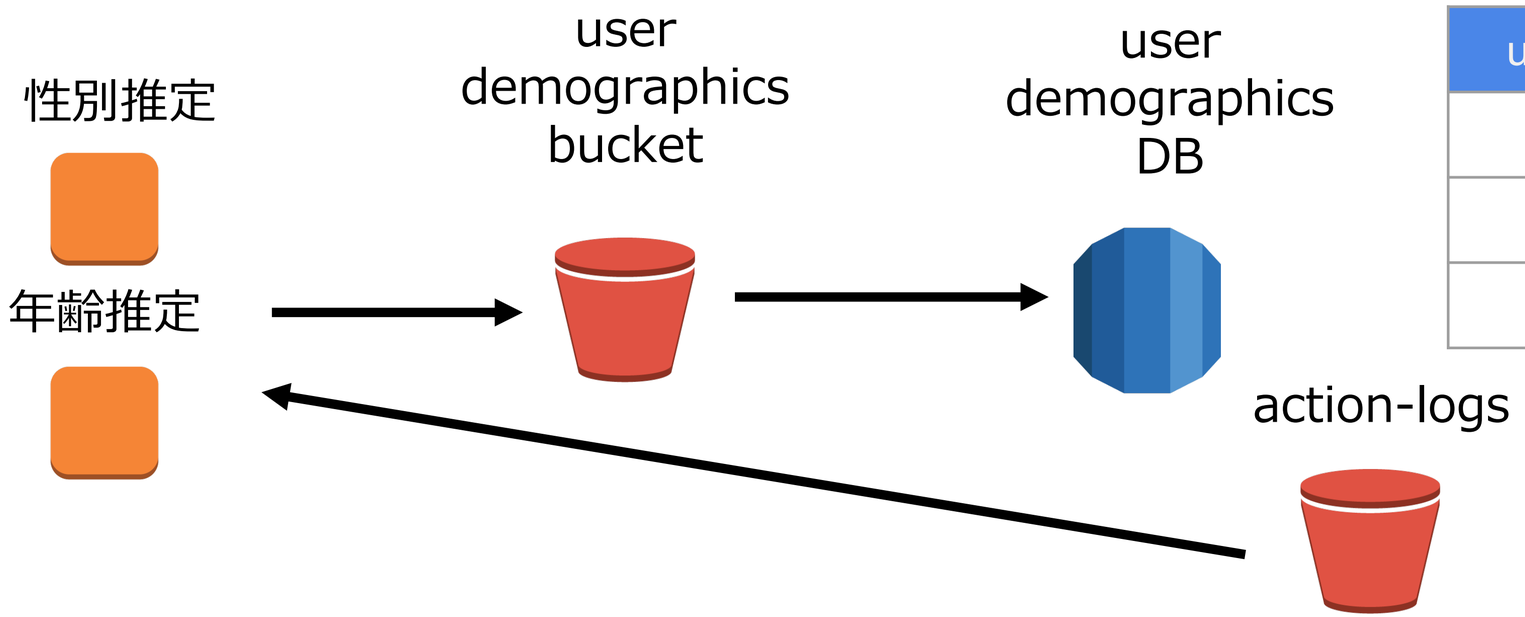

| user_id | gen_type |
|---|---|
| 100 | male |
| 101 | female |
| 102 | other |

# AWS上で動かしてみる (モニタリング)



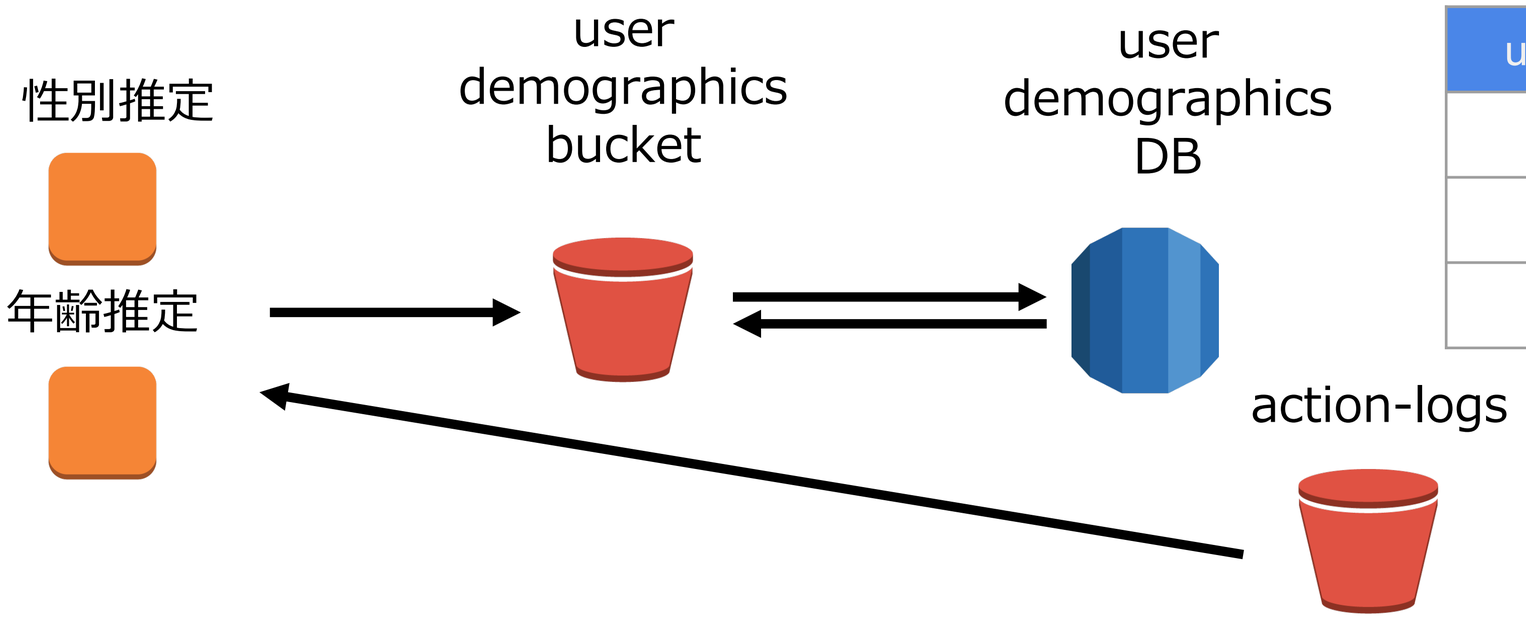

| user_id | gen_type |
|---|---|
| 100 | male |
| 101 | female |
| 102 | other |

# AWS上で動かしてみる (モニタリング)




| user_id | age_type |
|---|---|
| 100 | 20 |
| 101 | 30 |
| 102 | 50 |

# AWS上で動かしてみる (モニタリング)

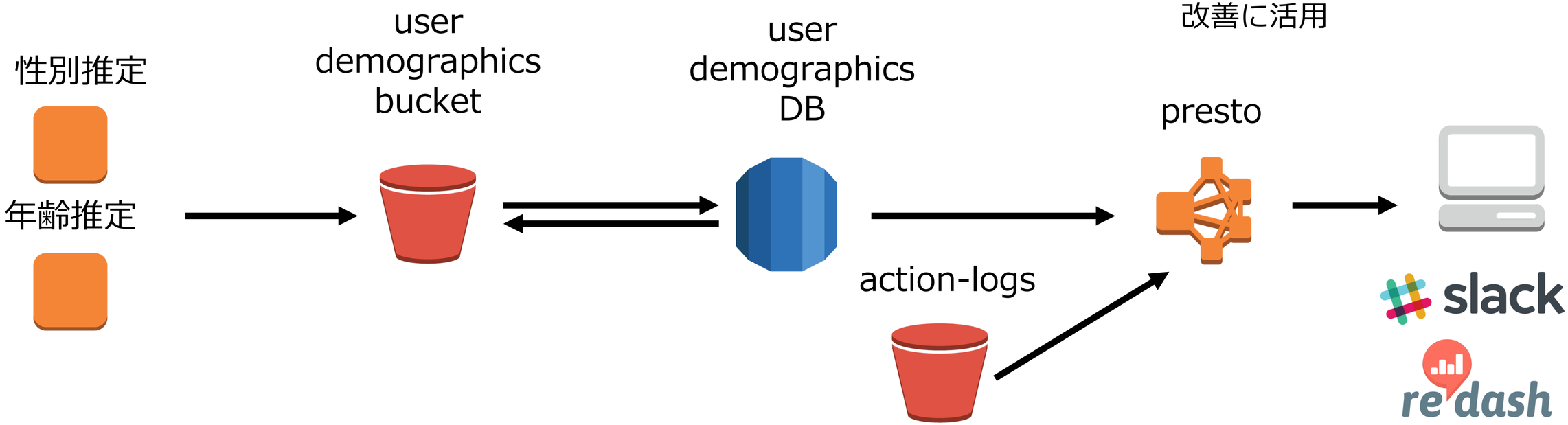



| user_id | gen_type | age_type |
|---|---|---|
| 100 | male | 20 |
| 101 | female | 30 |
| 102 | other | 50 |

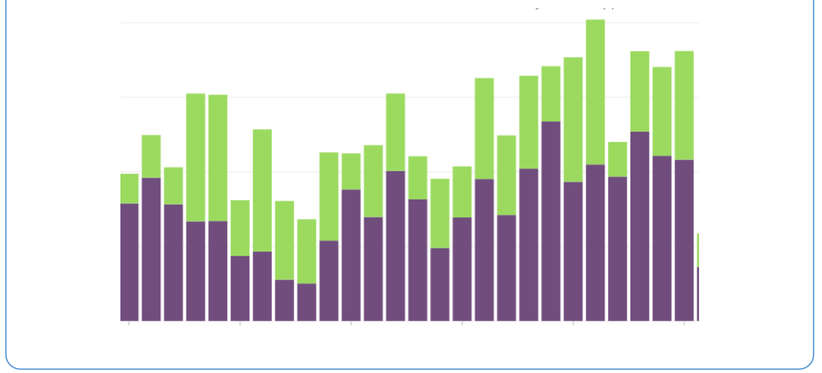
推定結果をJOINしたものをS3にも格納
(ストリーム処理で使用するため)

# AWS上で動かしてみる (モニタリング)

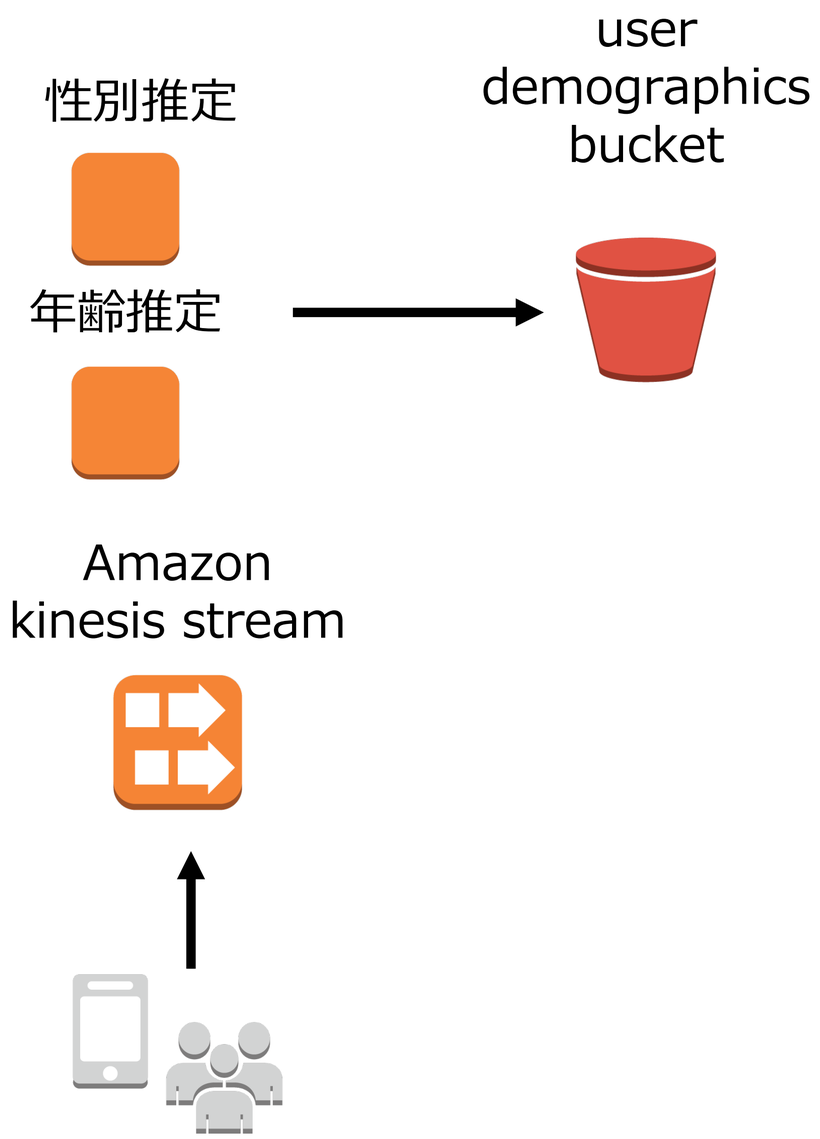
推定結果をRDBに格納し、日々の改善に活用




```
{"article_id": 12345, "user_id": 100, "action": "click"}
{"article_id": 12345, "user_id": 101, "action": "impression"}
```



# AWS上で動かしてみる (記事評価)

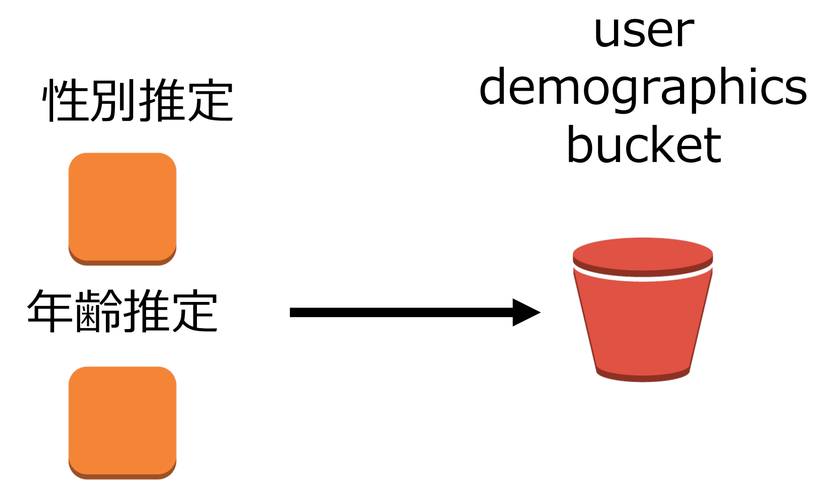




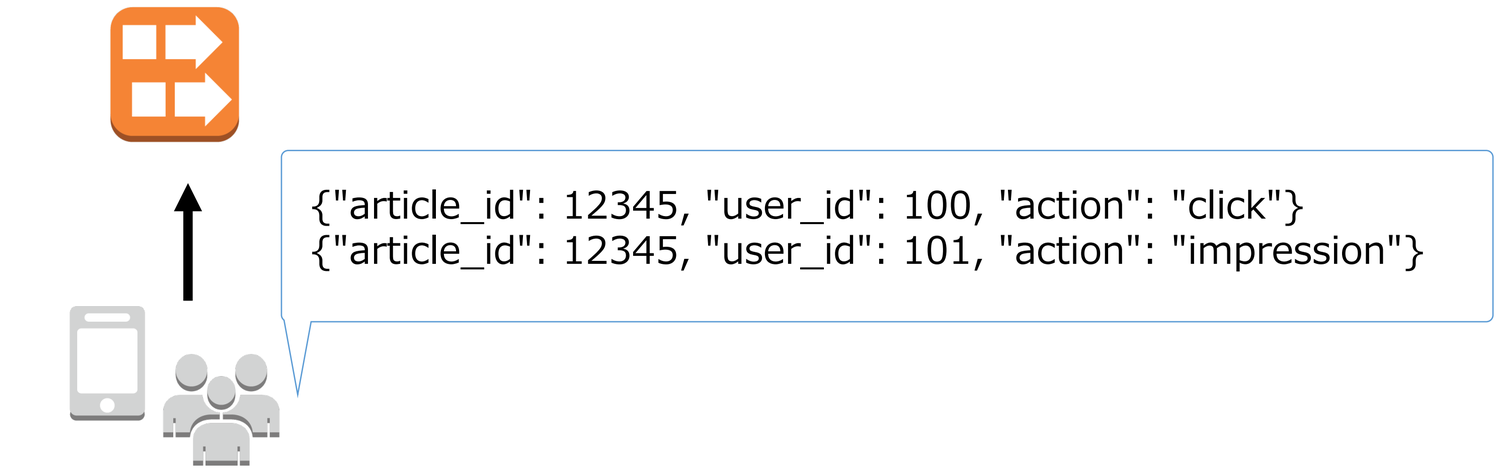


| user_id | gen_type |
|---|---|
| 100 | male |
| 101 | female |
| 102 | other |

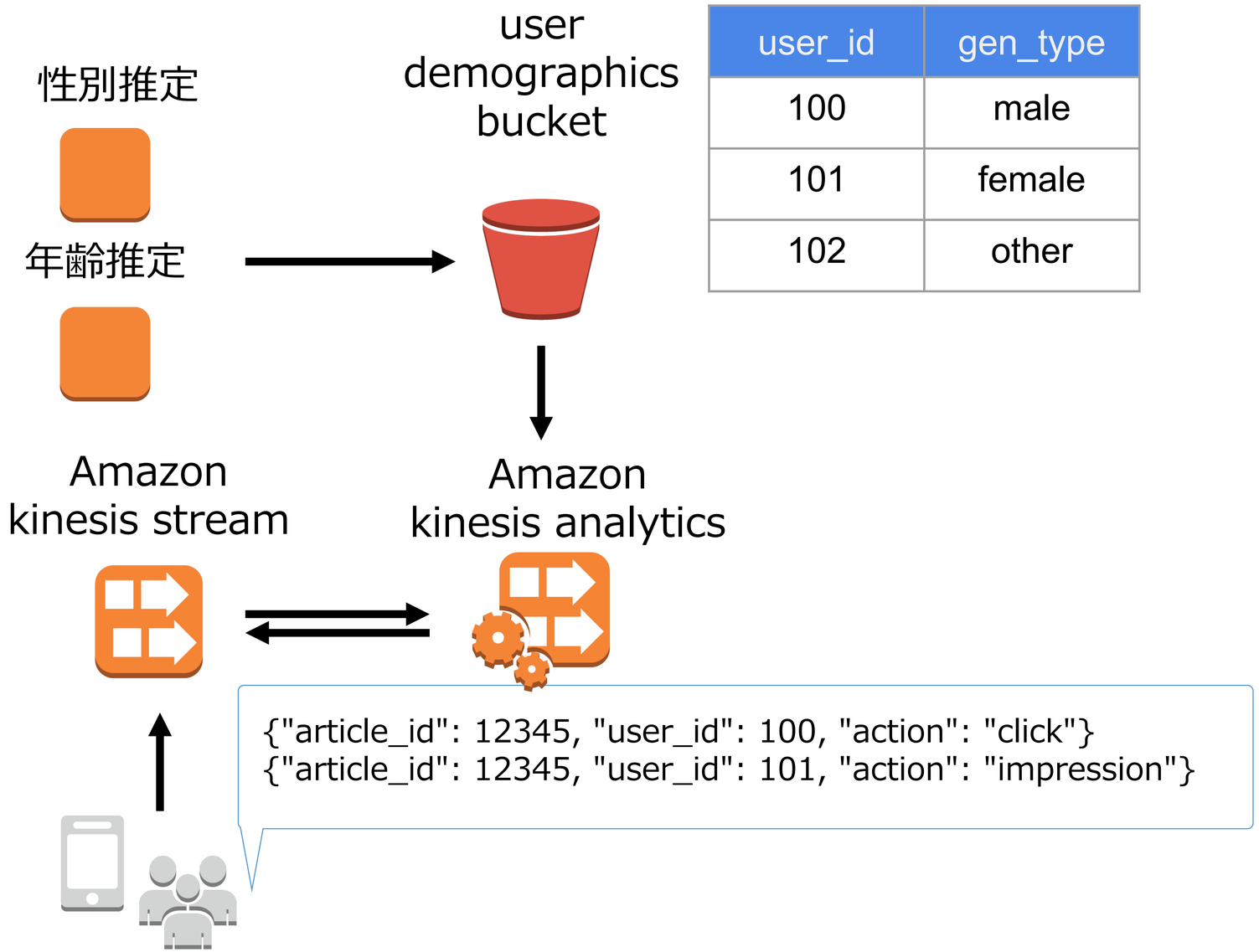
# AWS上で動かしてみる (記事評価)




| user_id | gen_type |
|---|---|
| 100 | male |
| 101 | female |
| 102 | other |



# AWS上で動かしてみる (記事評価)

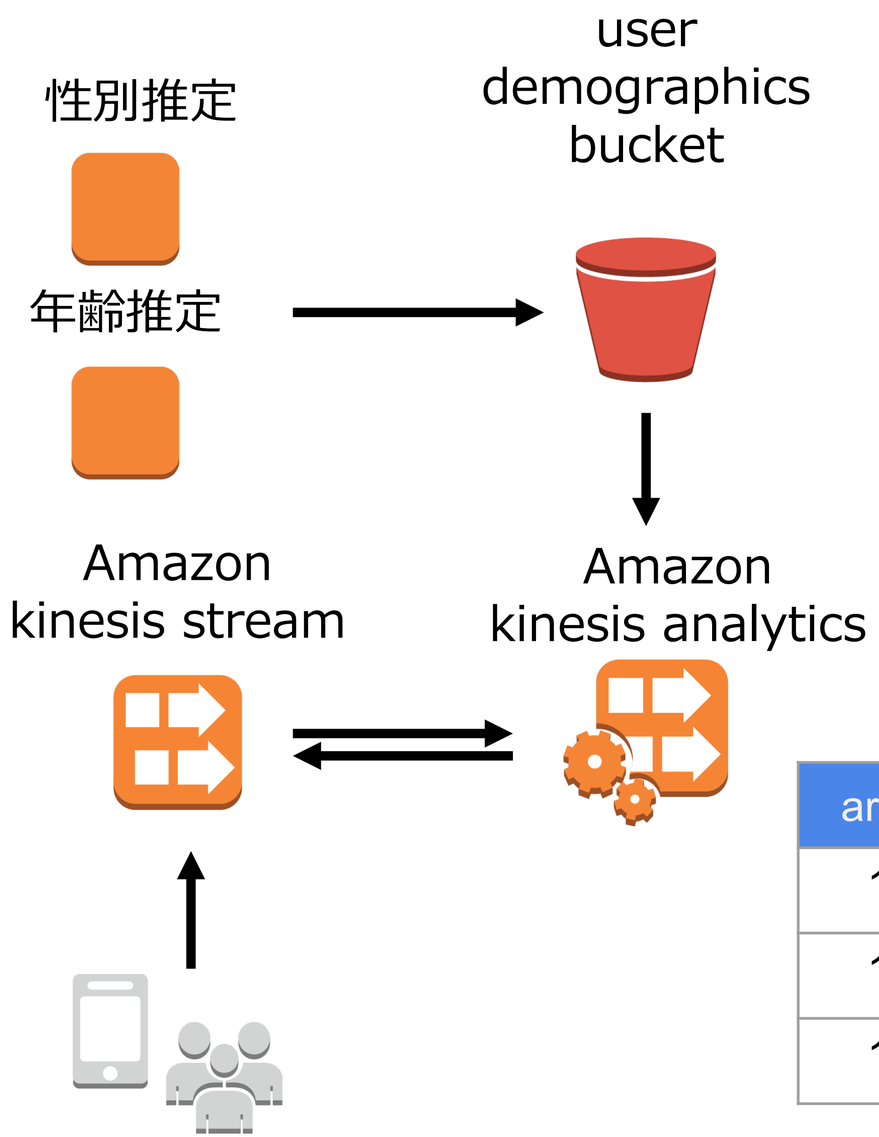



| user_id | gen_type |
|---|---|
| 100 | male |
| 101 | female |
| 102 | other |

kinesis streamのデータとS3のデータをkinesis analyticsでjoin

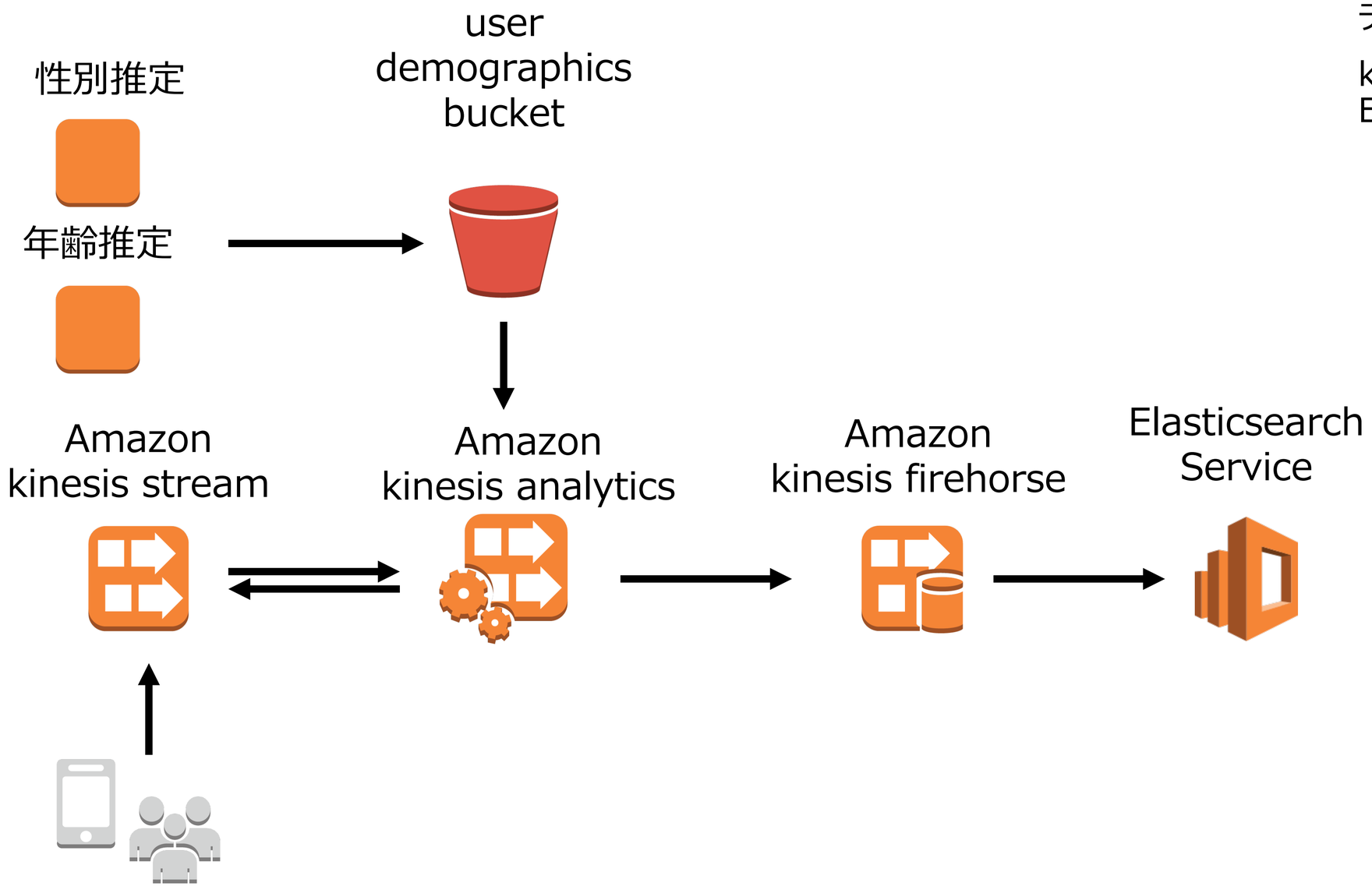
# AWS上で動かしてみる (記事評価)

kinesis streamのデータとS3のデータをkinesis analyticsでjoin




| article_id | gen_type | impression | click |
|---|---|---|---|
| 12345 | male | 10 | 4 |
| 12345 | female | 8 | 1 |
| 12346 | male | 3 | 1 |

# AWS上で動かしてみる (記事評価)




kinesis streamのデータとS3のデータをkinesis analyticsでjoin

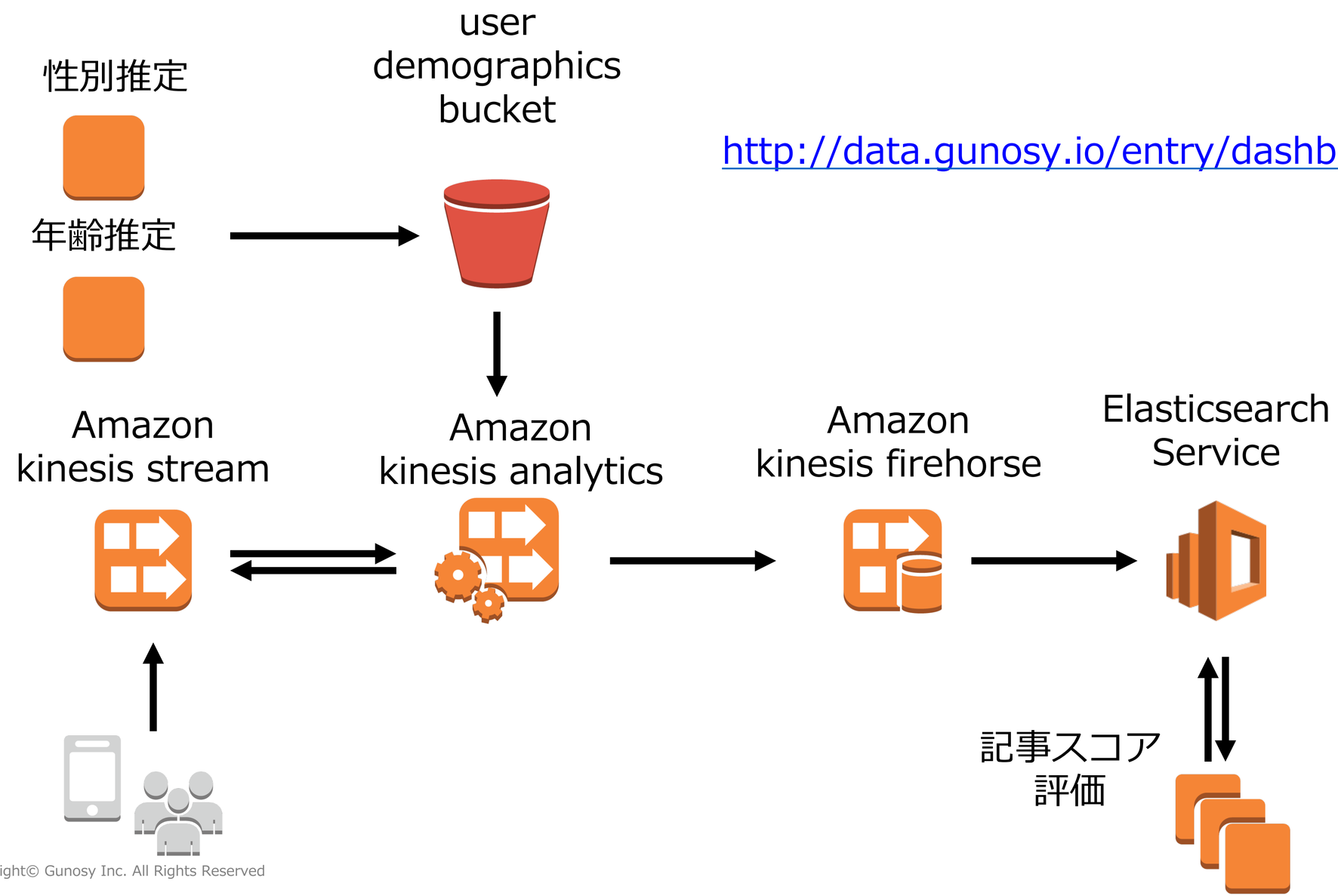
kinesis firehorseを通じてElasticsearchに格納

# AWS上で動かしてみる (記事評価)

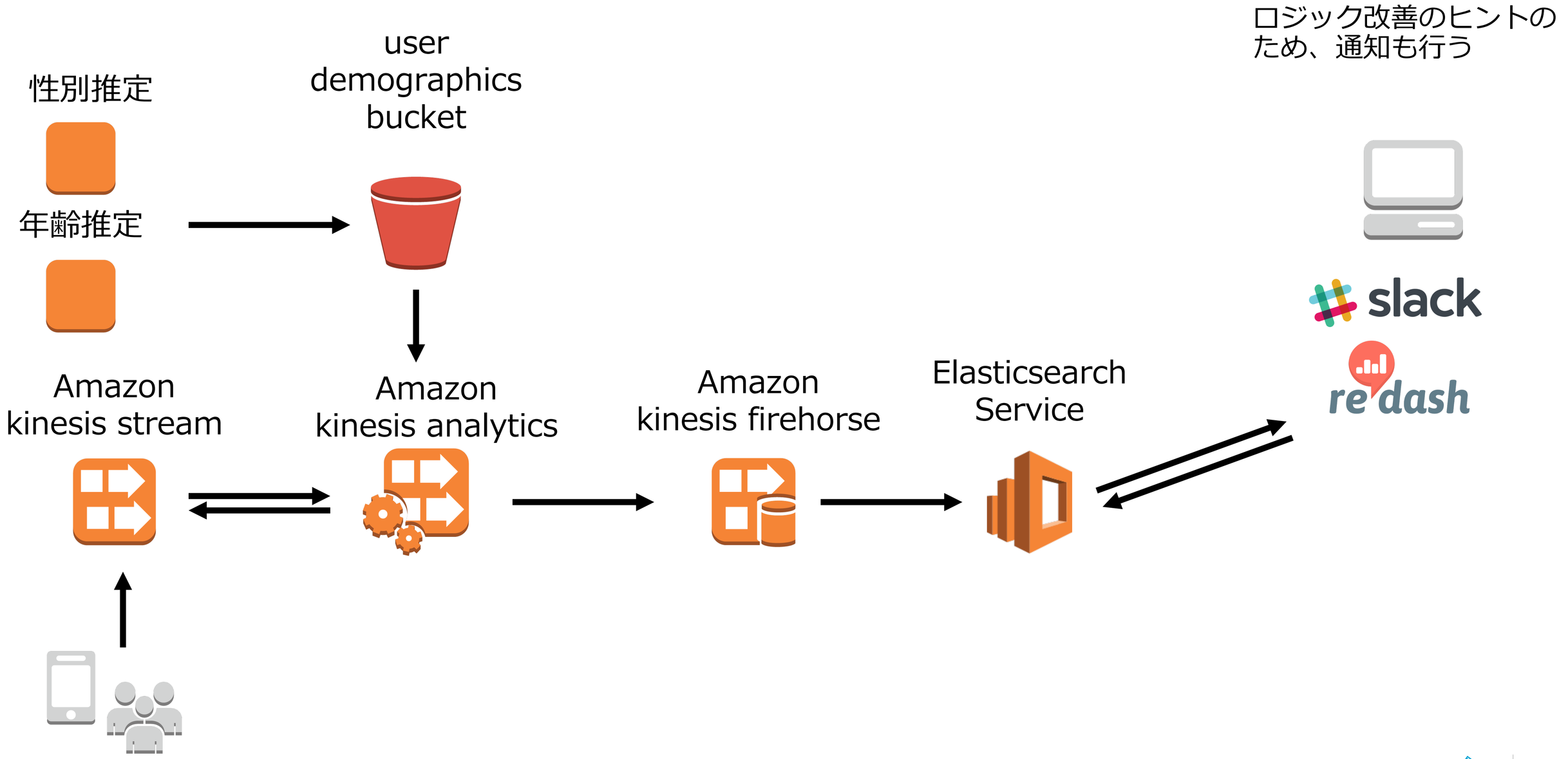
リアルタイム性が求められる
ニュース記事への評価に適用

詳しくはブログを参照

http://data.gunosy.io/entry/dashboard-with-kinesis-analytics



# AWS上で動かしてみる (リアルタイム通知 )



# 本日の発表の流れ

- Gunosyと機械学習
- 記事分類
- 属性推定 + スコアリング
- 効果測定 (ABテスト)

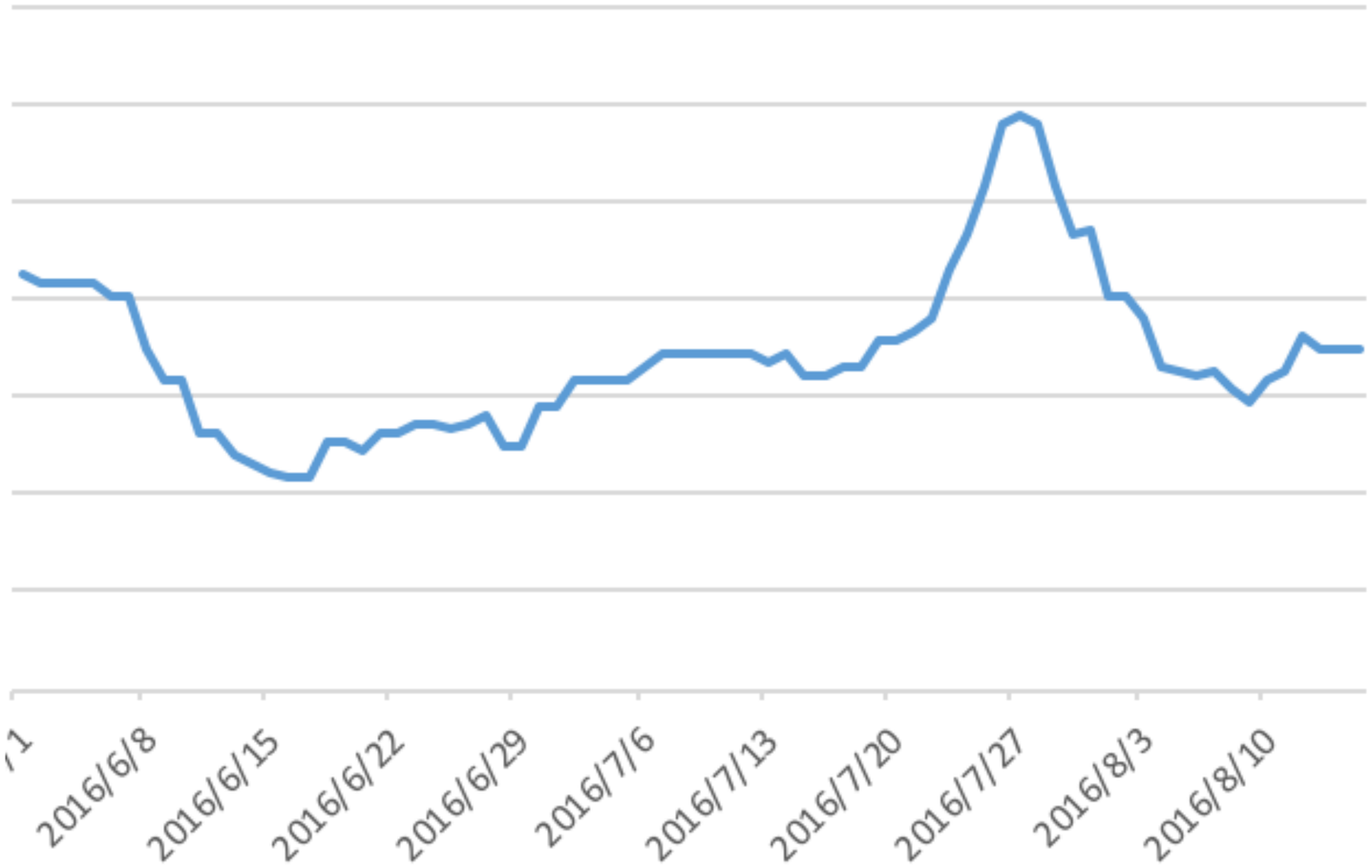
# 効果測定 (A/Bテスト)

多くの機械学習の効果測定タスクでは評価関数の最適化が目的

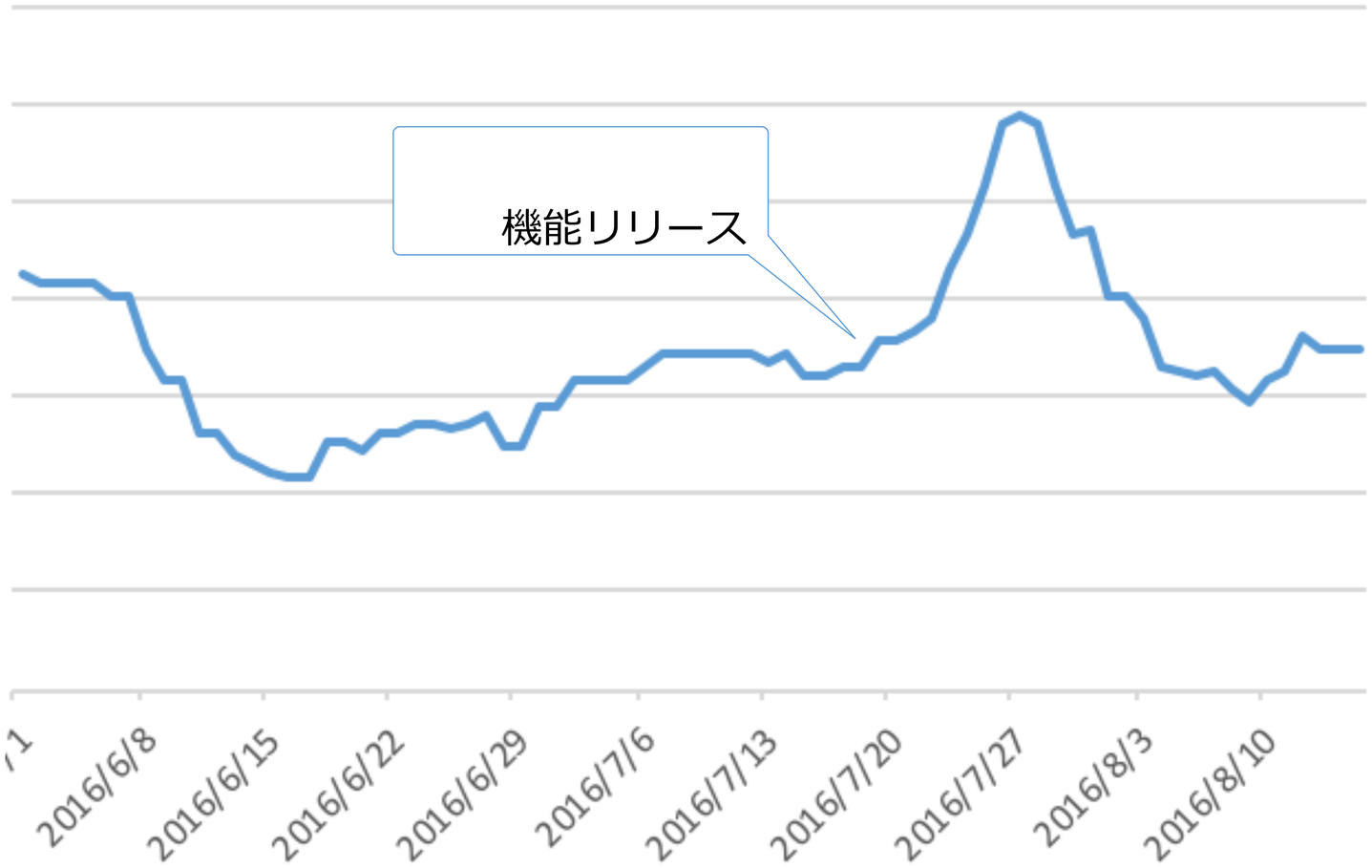
Gunosyが提供するサービスで最適化すべきはユーザの満足度

- 測定が難しい
- ニュースアプリのであるため、時流や季節要因などの影響を受けやすい
    - データに予期しないノイズが乗り、計測が困難になる
        - e.g 機能実装後にバグが発生し、実装により効果が上がったのか確認できない

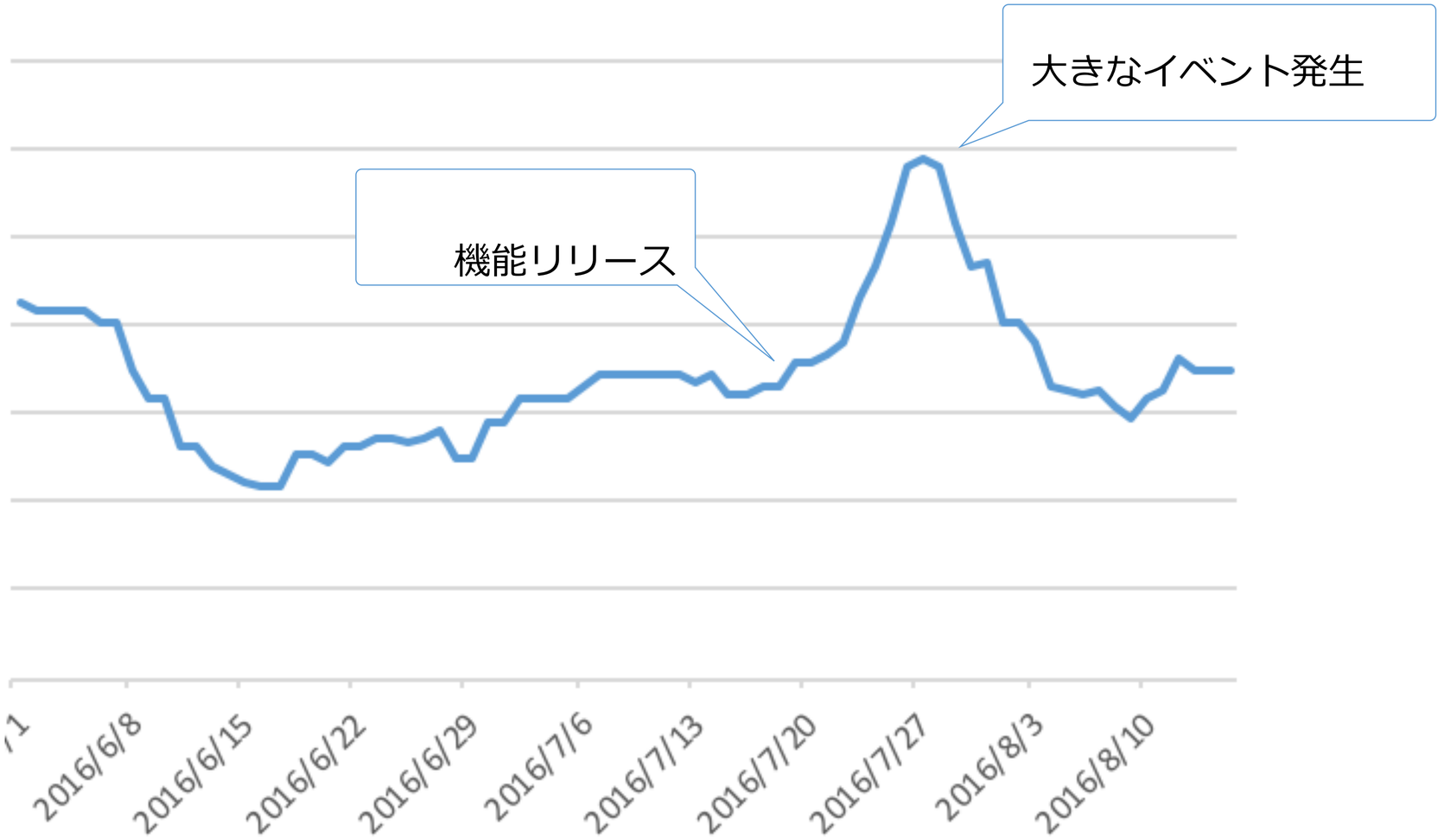
# 効果測定 (A/Bテスト) よくない例



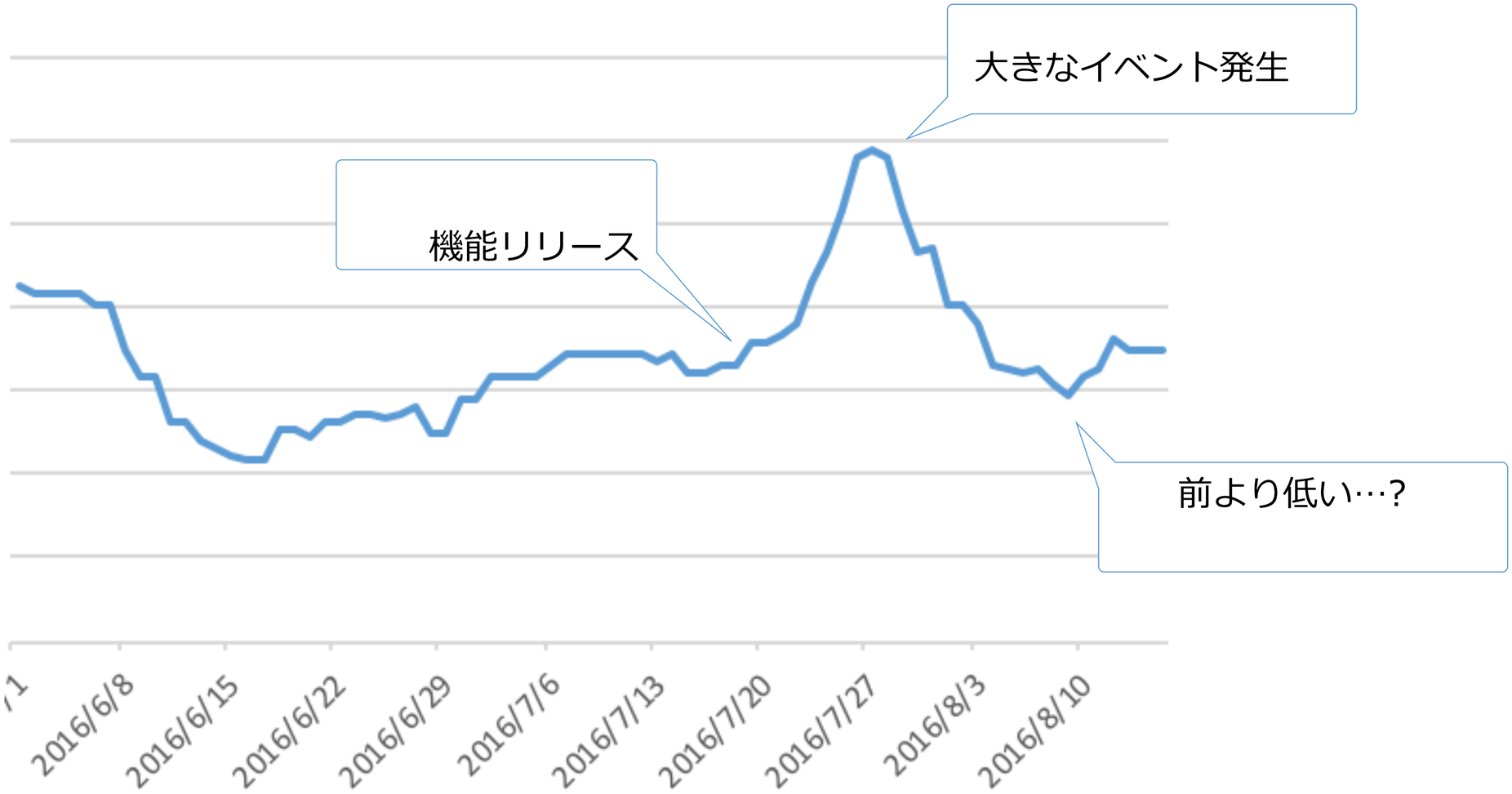
# 効果測定 (A/Bテスト) よくない例



# 効果測定 (A/Bテスト) よくない例



# 効果測定 (A/Bテスト) よくない例



# そこでA/Bテスト

特定のアルゴリズム(やUI)を2種類用意し、ユーザ毎に出し分けて検証する

メリット

- 時間変化などのノイズが入りにくい
- 最適化すべきメトリクスが決まってさえいれば単純なクロス集計で済む

# A/Bテストの例

特定のユーザ群にUIやアルゴリズムを出し分けるテストを行う

| | Test A | Test B |
|---|---|---|
| クリック率 | 5% | 6% |
| 滞在時間 | 30 | 35 |

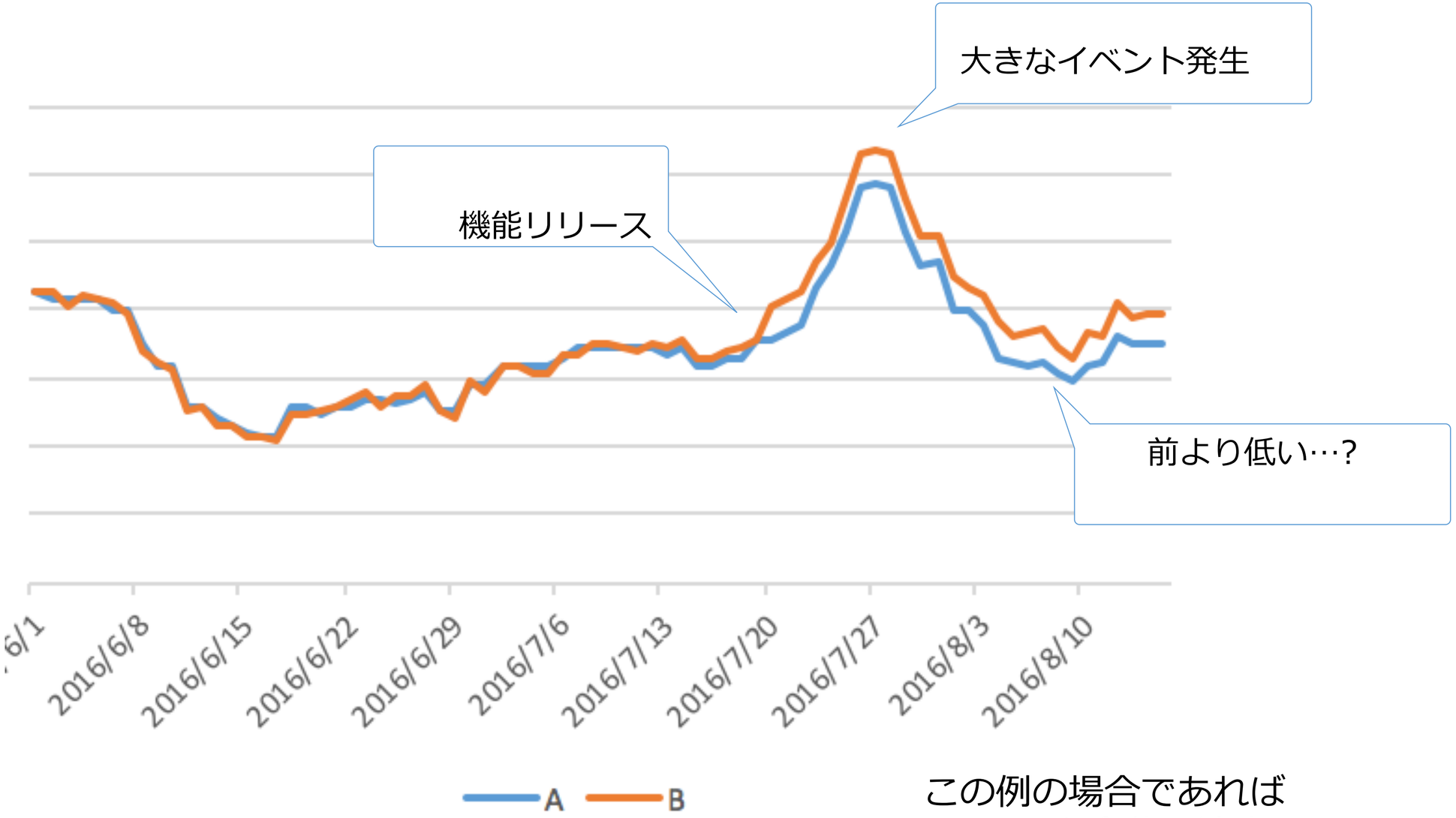
# A/Bテストの例

特定のユーザ群にUIやアルゴリズムを出し分けるテストを行う

| | Test A | Test B |
|---|---|---|
| クリック率 | 5% | 6% |
| 滞在時間 | 30 | 35 |

この例の場合であれば
Test Bを全体に適応
(※)実際は複数の指標を見ている

# A/Bテストの例




この例の場合であれば
Test Bを全体に適応
(※)実際は複数の指標を見ている

# A/Bテスト対象の選定

userのidを利用した割り当てを以前は利用

しかし割った値だと、テストが同時に複数走った場合に、
バイアスがかかる可能性がある

e.g. 同じ人がコントロールに当たる

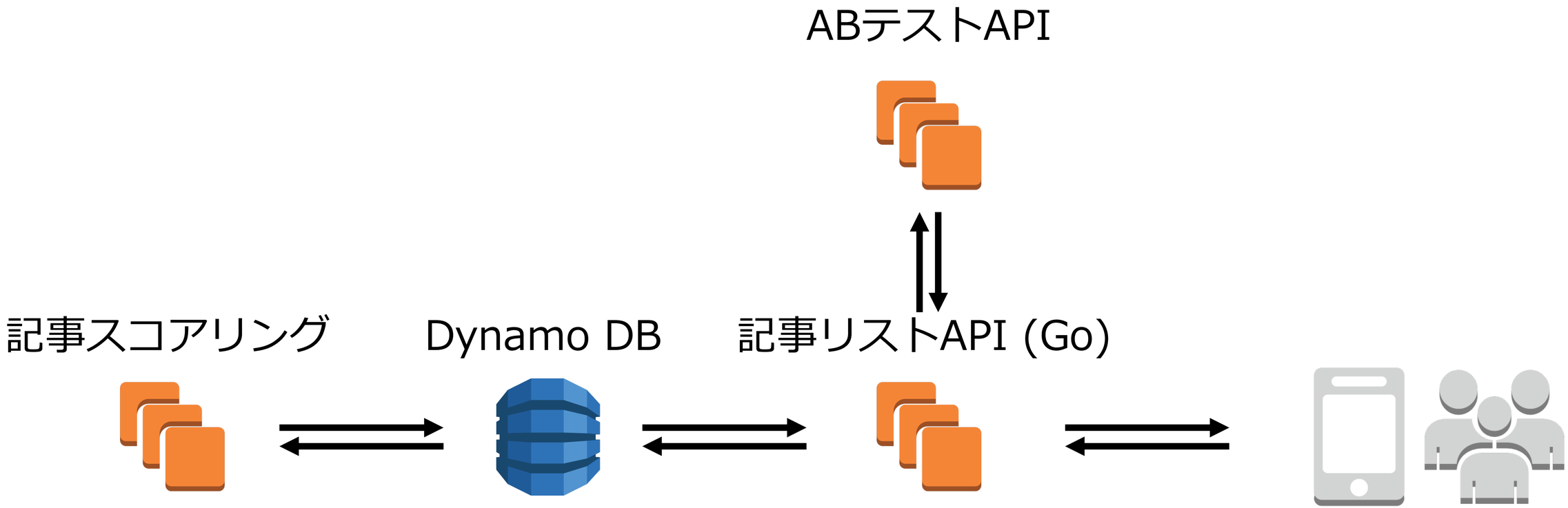
- 現在はハッシュ関数を利用した
  割り当てを行っている


詳細: http://data.gunosy.io/entry/ab_testing_assignment

# AWS上でのA/Bテスト実施

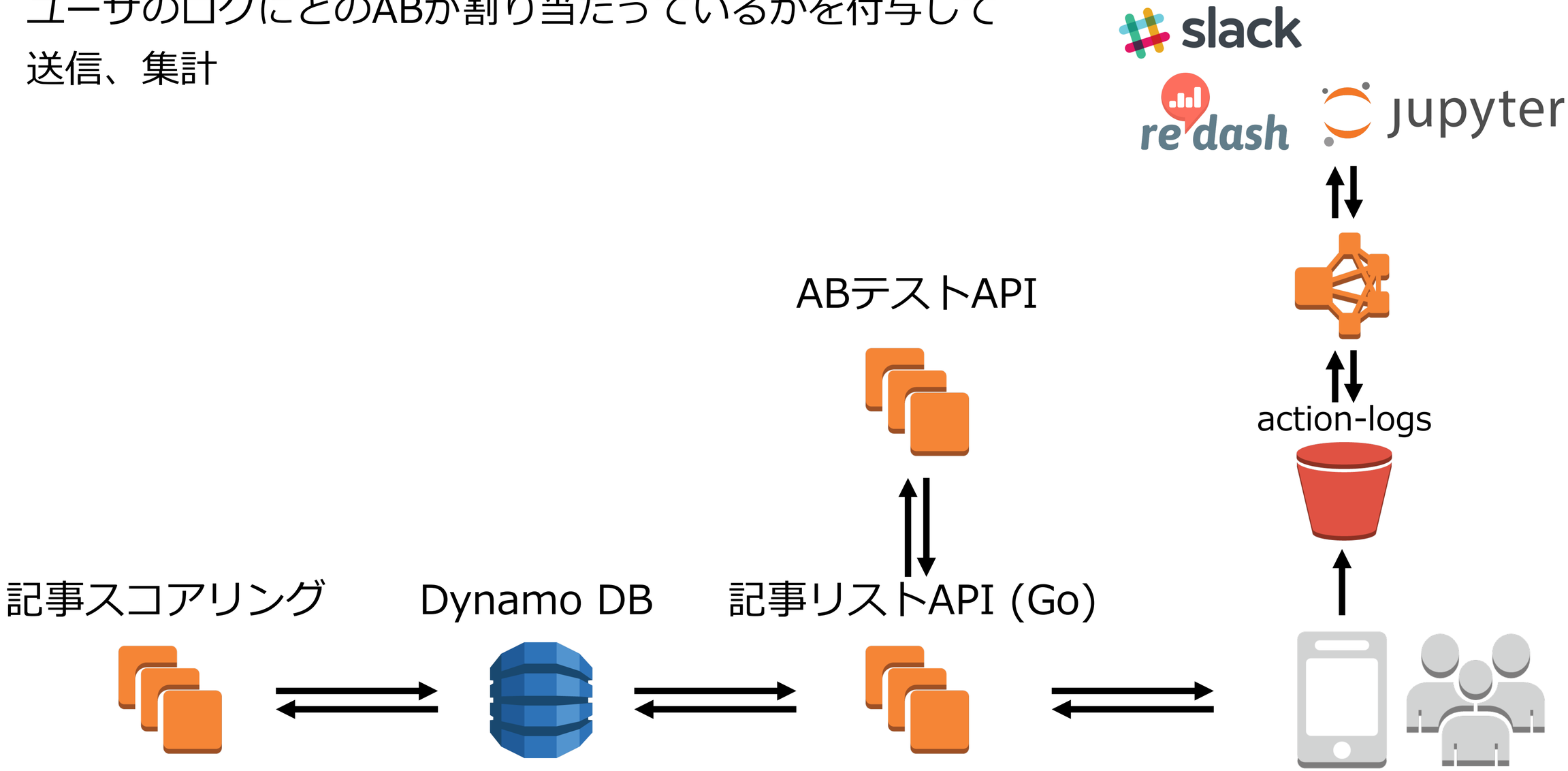
アルゴリズム毎にkeyを分けてDynamo DBに保存

ユーザに記事を配信するAPI(Go製)がDynamo DBに格納されたデータをユーザに返す



# AWS上でのA/Bテスト実施

ユーザのログにどのABが割り当たっているかを付与して
送信、集計

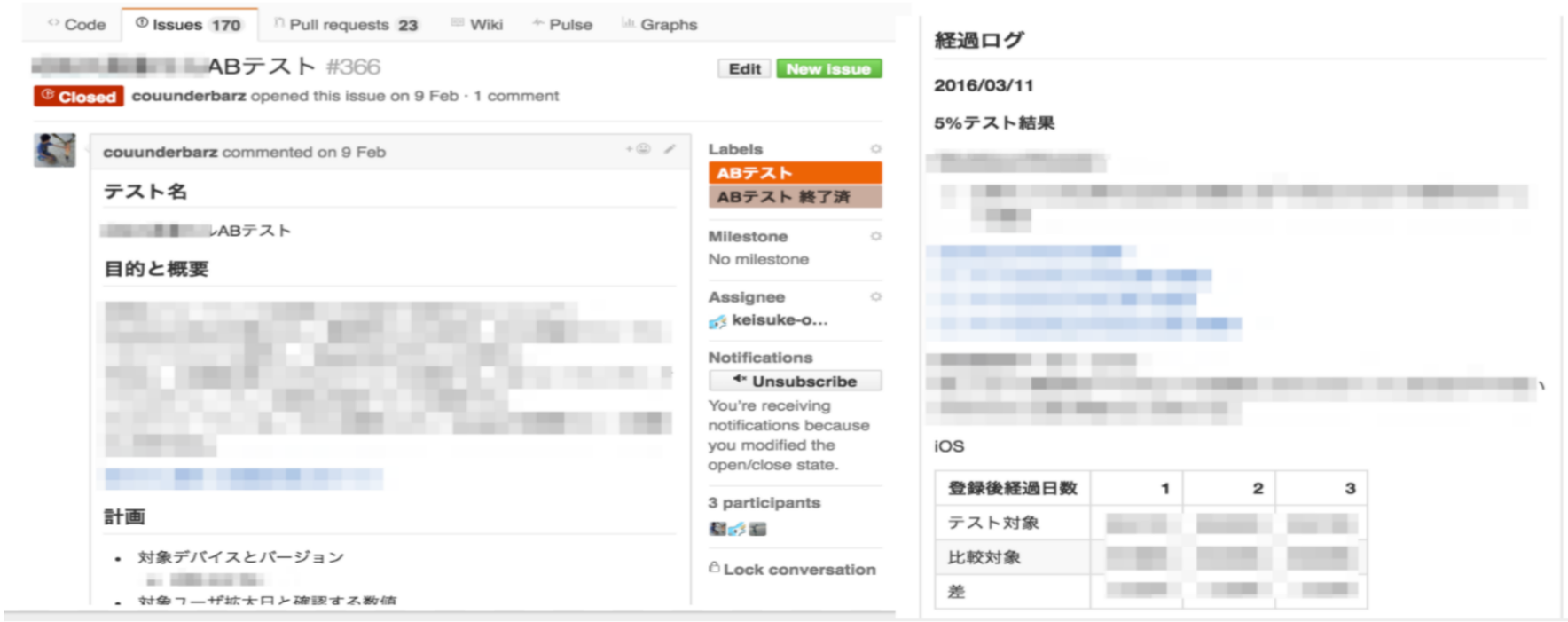


# 集計

ABテストごとに目的となる変数 (KPI) に影響がないかを確認

主にPython (Jupyter notebook)を用いて集計

Github上で集計コードを共通化



# 自動集計

Group Validation

ABテストが適切に割り当てられているかを確認、通知

自動化を進行中

# 伝えたかったこと

情報キュレーションサービスを対象に、AWS を用いて

人工知能技術を実サービスに応用する際の事例を紹介

機械学習ライブラリの充実により、ある程度の精度のもの(分類器などは)を作成するコストは下がっている

=> 実際のサービスで動かすことが重要

# おわりに

GunosyにおけるAWSを用いた機械学習の活用事例を紹介

- 記事分類
- 属性推定 + スコアリング
- 効果検証

今後はオンラインでの推薦や、より高度な自然言語処理、
画像処理などにもチャレンジしていきたい

# その他

**「Gunosy データ分析ブログ」**

http://data.gunosy.io/

→データに基づいたプロダクト改善のためのブログ

**「Gunosy データマイニング勉強会」**

https://gunosy-dm.connpass.com/

→ 輪読 + 論文紹介する会

# 人材募集中です

**機械学習・自然言語処理エンジニア** https://www.wantedly.com/projects/83871

**データ分析エンジニア** https://www.wantedly.com/projects/83864

**データプラットフォームエンジニア** https://www.wantedly.com/projects/95424



ご静聴ありがとうございました