

マイマガジンシステムでの SageMakerを活用したレコメンドシステム構築事例

自己紹介

秦 将之

所属 サービスデザイン部

- 略歴
- ・ 某社でインフラエンジニア
 - ・ 某社でアプリケーション開発
 - ・ 2010年からドコモ入社
 - ・ ビジネス部を経てサービスサーバ開発へ

基本的にインフラ、アプリ、ML何でも見ます！



アジェンダ

- **マイマガジンとは**
- **既存レコメンドロジックが抱える課題**
- **SageMakerの導入による既存課題の解決**
- **導入効果**
- **まとめ（導入メリット等に関する考察）**

会社紹介

株式会社NTTドコモ

通信事業だけでなく様々なスマートライフ事業を展開



マイマガジンについて

マイマガジンは2013年からスタート！

主にLIVE UXで豊富なジャンルのニュースを届けるキュレーションアプリ



話題のジャンル



オリジナルジャンル



クーポン・マンガ

最近ではらくらくスマートフォン版等をリリース！

マイマガジンのシステム規模



- ジャンル数は**34**（2020年3月時点）
- トランザクションは平常時**3,000~5,000TPS**
- ピーク時は**7,000TPS**
- アクセス数は**約2.4億／日**
- AWSで稼働中
- サーバ台数は商用・検証合わせて**約320台**稼働（Fargate含む）

マイマガジンの記事レコメンド

ユーザの趣味・趣向に合わせた
記事の表示



ユーザーが見たい記事を出す⇒ユーザーの訪問回数が増える

**マイマガジンのメディア価値が向上
最終的なKGI達成につながる（売上貢献）**

今回は、SageMakerを使って 如何にしてユーザーが見たい記事を出すよう改善したか お話しします！

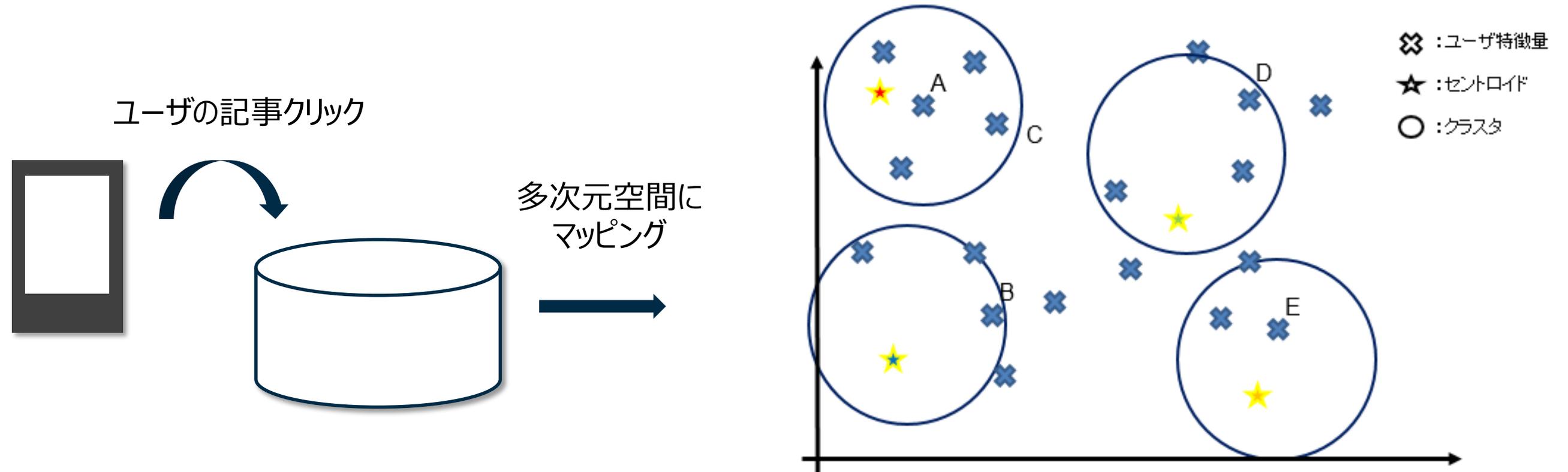


記事レコメンドが抱える問題

マイマガジンのレコメンドロジック

① ユーザクラスタ生成

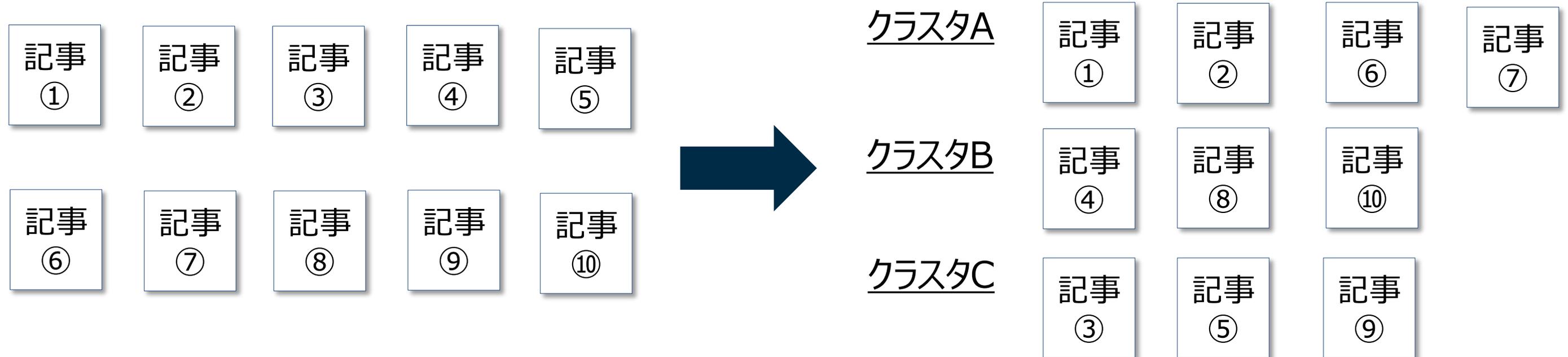
ユーザの記事クリック傾向から、独自に特徴量ベクトルを生成しクラスタ化



マイマガジンのレコメンドロジック

②レコメンド記事推論

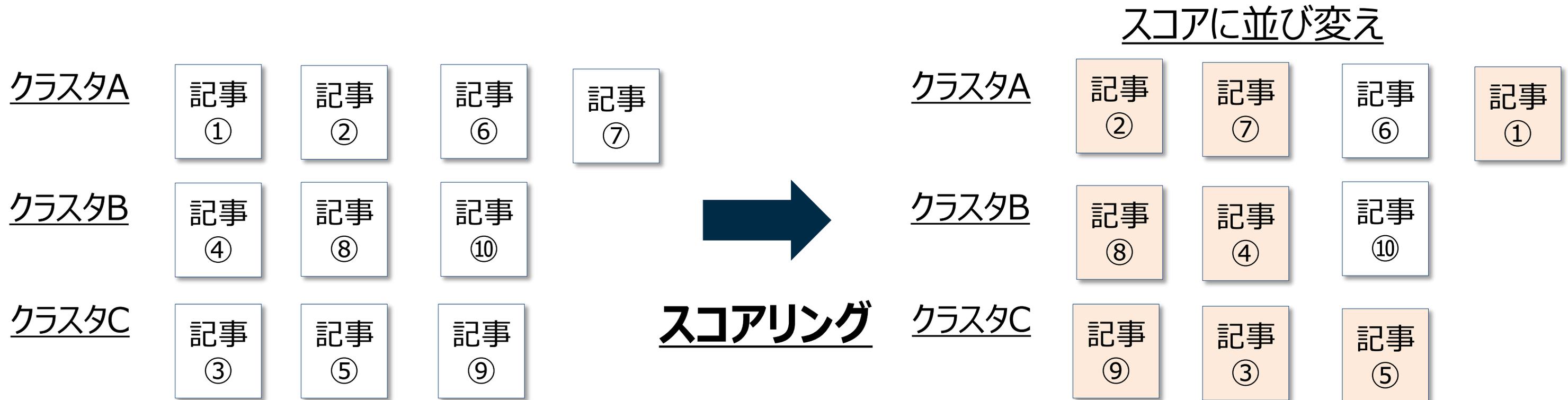
クラスタセントロイドと記事コンテンツ特徴量から各クラスタに記事を分類



マイマガジンのレコメンドロジック

③レコメンド記事のスコアリング

分類された記事を独自の数式でスコア化しておすすめ優先度を定める



仮説

スコアが高い記事ほど、ユーザがクリックしてくれるはず

じゃあ、その仮説は正しいのか？

環境の変化

- ユーザ数がここ3年で**1.36倍**
- ジャンルはここ一年で**約15個オープン**
- ジャンルの増加に伴い掲載**メディアも増加傾向**
- 様々なサービスロジックを追加
- ユーザの**利用傾向は少しずつ変化、興味も多様化**



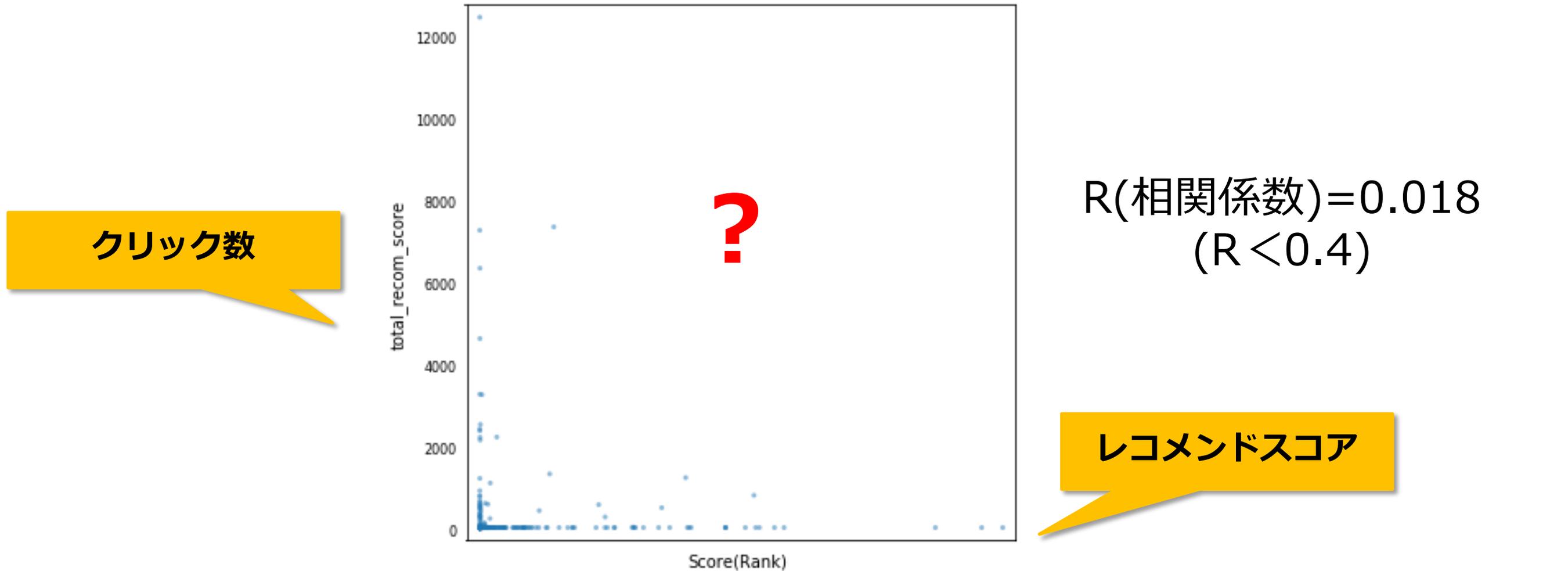
※2020/3/20時点のジャンル開設お知らせ

ジャンルによってはレコメンド記事のクリックが伸びていなかった！

以前は正しかったのだけど・・・

じゃあ、その仮説は正しいのか？

仮説通りならスコアが高いとクリック数も高い（正の相関）はず！？



相関がない！レコメンド記事も統一性がない(野球記事が多いとか)

学習ロジックがサービス性に合わなくなって来ている？
何が課題なのか？

学習に関する課題

学習モデル自体の課題

- ✓ レコメンド記事推論に教師なし学習を活用している
- ✓ チューニングをクリック数の改善につなげるのが難しい

環境面での課題

- ✓ 複数回のイテレーションが施行できない（性能）
- ✓ 独自実装なので色々試せない
- ✓ 各ロール（役割）が適切に機能していない

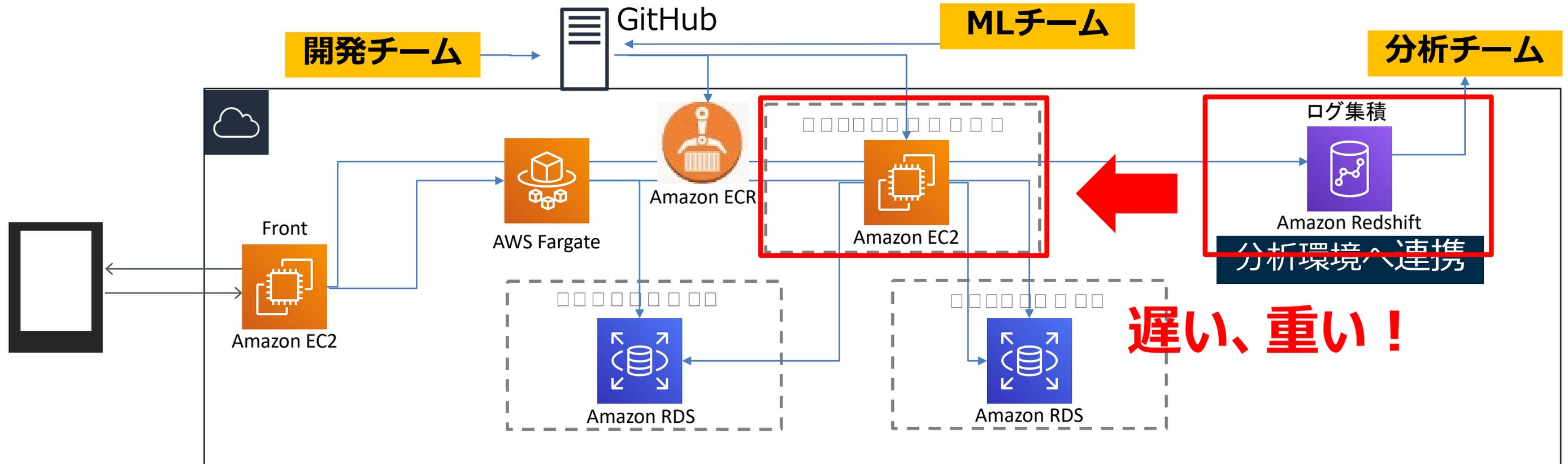
特に環境面課題解決は、アーキテクチャの見直しも必須！

アーキテクチャを見てみると

処理が並列化できず処理性能がEC2のスペックに依る

分析対象データ（ログ・記事）増 = 処理時間の指数関数的な増

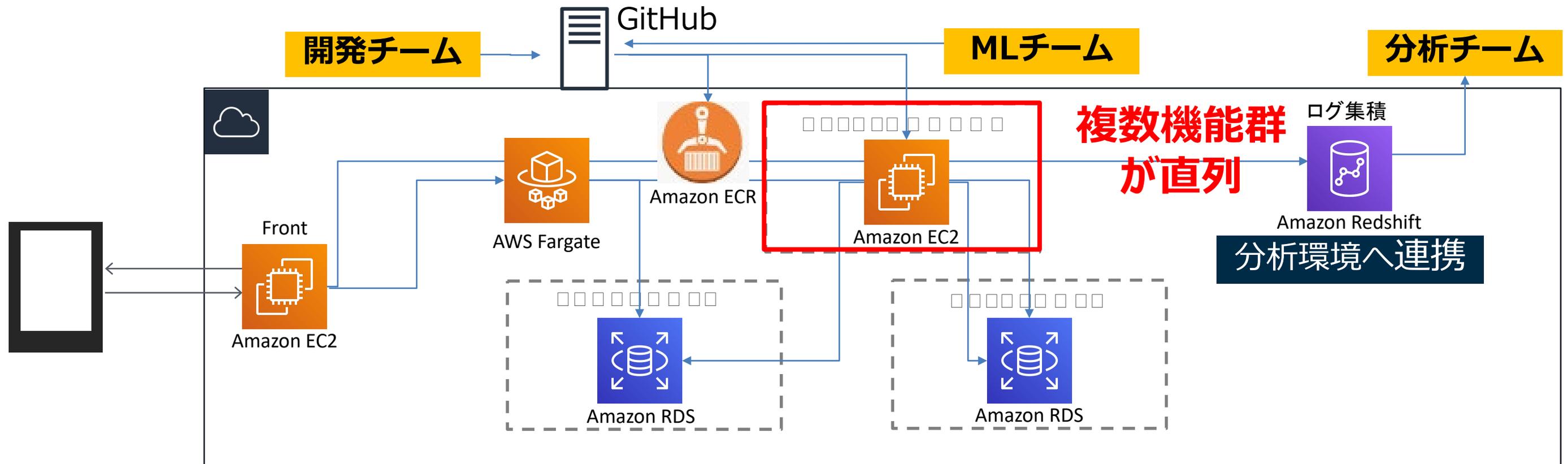
※ユーザログは日に数億、記事は日に数千のオーダー



せっかくいいアルゴリズム入れてもデータ量で性能がネックになる

アーキテクチャを見てみると

独自の組込実装で**アルゴリズム変更やチューニングに大規模な改修が必要**
機能が密結合な為、各チームが**自由に学習環境を触れない**



各ロールのトライ&エラーが重要⇒環境制約上×

目指すべき姿

レコメンド記事クリック数を向上 = ユーザのマイマガ利用 ↑
ユーザ満足度向上が、メディア価値を上げKGI達成へつながる

- **教師あり学習アルゴリズムの採用（独自計算ロジックの脱却）**
- **アーキテクチャの刷新**
 - ✓ スケーリング容易
 - ✓ アルゴリズムやハイパラを色々試せる
 - ✓ 各ロールが連携しあいながら機能疎結合に活動

そこでSageMakerと出会う



SageMakerに期待したこと

- **教師あり学習アルゴリズムの採用**
⇒ **アルゴリズムの扱いやすさ（ビルトインアルゴリズムの利用）**
- **アーキテクチャの刷新**
 - ✓ スケーリング容易
⇒ **サーバレス（後述するがコストも削減）**
 - ✓ アルゴリズムやハイパラを色々試せる
⇒ **豊富なビルトインアルゴリズム、チューニングの容易性**
 - ✓ 各ロールが連携しあいながら機能疎結合に活動
⇒ **各ロールが扱う機能を疎結合に設計可能**

SageMakerに期待したこと

- **教師あり学習アルゴリズムの採用**
⇒ **アルゴリズムの扱いやすさ（ビルトインアルゴリズムの利用）**
- **アーキテクチャの刷新**
 - ✓ スケーリング容易
⇒ **サーバレス（後述するがコストも削減）**
 - ✓ アルゴリズムやハイパラを色々試せる
⇒ **豊富なビルトインアルゴリズム、チューニングの容易性**
 - ✓ 各ロールが連携しあいながら機能疎結合に活動
⇒ **各ロールが扱う機能を疎結合に設計可能**

そもそもどのようなアルゴリズムがよかったか

入力特徴量へ記事クリック数への寄与度が高い特徴量の組合せを複数組込める。
(交互作用)

⇒ユーザの利用形態や趣味趣向が複雑化しても特徴量の組合せを増やせる

■ 入力特徴量例

ユーザの特徴	記事の特徴	
ユーザ特徴量	コンテンツ特徴量	コンテキスト等
01000002	010101... 23 1	2020320
00000103	011101... 23 1	2020410
00100004	000101... 23 1	2020325
00010005	110001... 23 1	2020505
10000006	010001... 23 1	2020321
01000007	010100... 23 1	2020419

必要に応じて
特徴量を追加

そもそもどのようなアルゴリズムがよかったか

入力特徴量へ記事クリック数への寄与度が高い特徴量の組合せを複数組込める。
(交互作用)

⇒ユーザの利用形態や趣味趣向が複雑化しても特徴量の組合せを増やせる

■ 入力特徴量例

ユーザの特徴	記事の特徴	
ユーザ特徴量	コンテンツ特徴量	コンテキスト等
01000002	010101...	23 1 2020320
00000103	011101...	23 1 2020410
00100004	000101...	23 1 2020325
00010005	11000	
10000006	01000	
01000007	01010	

必要に応じて
特徴量を追加

豊富な組み込みアルゴリズムから
いくつかアルゴリズムを検討してみて
Factorization Machinesを採用

(記事)コンテンツ特徴量の事前処理の工夫

既存ロジックの、記事コンテンツ特徴量ベクトルは**本文の形態素から生成**

- ✓ スパースな特徴量が余りにも多い (FMはスパースなデータを扱いやすいが)
- ✓ 次元が大きすぎる (次元の呪い)

スパースの回避と次元数圧縮のため**BERT**を導入しコンテンツ特徴量を表現

SageMakerを動かしてみる

SageMakerの利用

SageMakerのビルトインアルゴリズムの事前検証として
商用データを用いたハイパーパラメータチューニングを実施

- ①ビルトインアルゴリズムでベースラインモデルの構築
- ②ハイパーパラメータの自動最適化
- ③評価データの推論（過学習となっていないか）



① ビルトインアルゴリズムでベースラインモデルの構築

ベースラインとなる学習モデルを決める際
すべてのハイパーパラメータを見ようとするとかかなり辛い

ビルトインアルゴリズムでプリセットのハイパーパラメータを利用して
チューニング対象パラメータをスコープ、ベースラインモデルを生成！

パラメータ名	説明
feature_dim	入力特徴空間の次元。これはスパース入力では非常に高くなる場合があります。 必須 有効な値: 正の整数。推奨値の範囲: [10000,10000000]
num_factors	因数分解の次元。 必須 有効な値: 正の整数。推奨される値の範囲: [2,1000]、通常は 64 が最適です。
predictor_type	予測子のタイプ。 <ul style="list-style-type: none">binary_classifier: 二項分類タスクの場合。regressor: 回帰タスクの場合。 必須 有効な値: 文字列: binary_classifier または regressor
bias_init_method	バイアス項の初期化方法。 <ul style="list-style-type: none">normal: 平均が 0 で標準偏差が bias_init_sigma で指定された正規分布からサンプリングしたランダム値で重みを初期化します。uniform: [-bias_init_scale, +bias_init_scale] で指定された範囲から均一にサンプリングされたランダム値で重みを初期化します。

②ハイパーパラメータの自動最適化

変更対象のハイパーパラメータが見えてきたら後は試行錯誤
下記のようにオブジェクトにパラメータ設定して複数回学習を実施

学習を動かしたら目的メトリクス値を確認して最大となっている
ハイパーパラメータの組み合わせを見つける。

```
hyperparameter_ranges={'epochs': IntegerParameter(1, 200),  
                        'mini_batch_size': CategoricalParameter([8, 16, 32, 64, 128, 256, 512, 1024, 2048])  
                        }  
objective_metric_name='test:binary_classification_accuracy'
```

名前	ステータス	目標メトリクス値	作成時刻	トレーニング期間
factorization-machin-200312-0522-100-	Completed	9.112876892089844	Mar 12, 2020 06:01 UTC	3分
factorization-machin-200312-0522-099-	Completed	9.21034049987793	Mar 12, 2020 06:00 UTC	2分
factorization-machin-200312-0522-098-	Completed	8.430628776550293	Mar 12, 2020 06:00 UTC	2分
factorization-machin-200312-0522-097-	Completed	9.307804107666016	Mar 12, 2020 05:59 UTC	2分
factorization-machin-200312-0522-096-	Completed	8.186969757080078	Mar 12, 2020 05:59 UTC	2分
factorization-machin-200312-0522-095-	Completed	7.943309783935547	Mar 12, 2020 05:58 UTC	2分
factorization-machin-200312-0522-094-	Completed	8.771753311157227	Mar 12, 2020 05:58 UTC	2分

ハイパラをrange指定すると
その範囲で探索可能

③ 評価データの推論

評価用データを推論を実施し、過学習になっていないか確認
併せてレコメンドスコアとクリック数の相関を分析（後述）

最適なハイパーパラメータは実際に商用運用してみるとサービスの変化
に変わる可能性もあるので定期的な見直しが必要
それ前提で学習しやすいアーキテクチャであることはもっと重要

どのようなアーキテクチャにしたか

SageMaker導入によるアーキテクチャの刷新

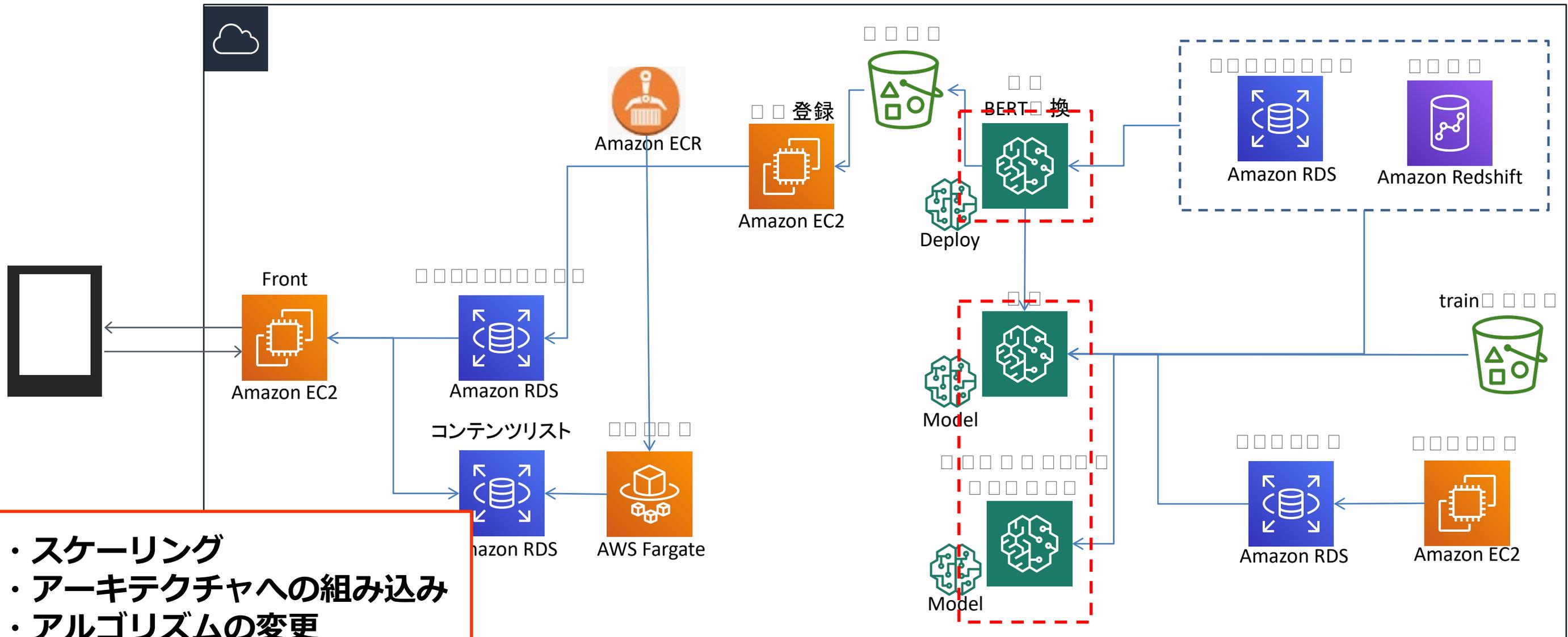
- 教師あり学習アルゴリズムの採用
⇒アルゴリズムの扱いやすさ（ビルトインアルゴリズムの利用）

➤ アーキテクチャの刷新

- ✓ スケーリング容易
⇒サーバレス（後述するがコストも削減）
- ✓ アルゴリズムやハイパラを色々試せる
⇒豊富なビルトインアルゴリズム、チューニングの容易性
- ✓ 各ロールが連携しあいながら機能疎結合に活動
⇒各ロールが扱う機能を疎結合に設計可能

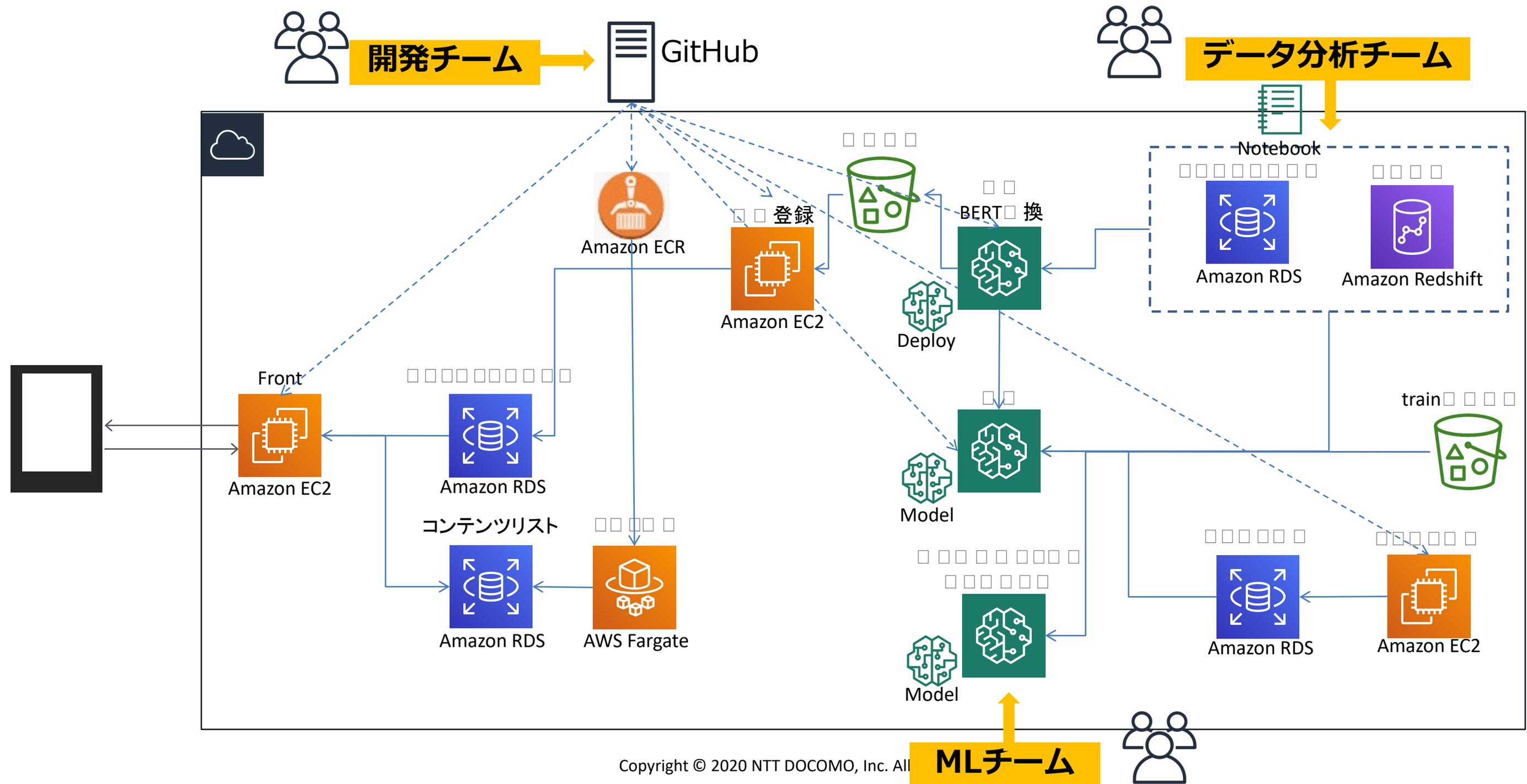
SageMaker導入のアーキテクチャ

AP開発,ML,データ分析チーム各々の活動が他チームの妨げにならないよう
学習機能、推論機能を疎結合に設計

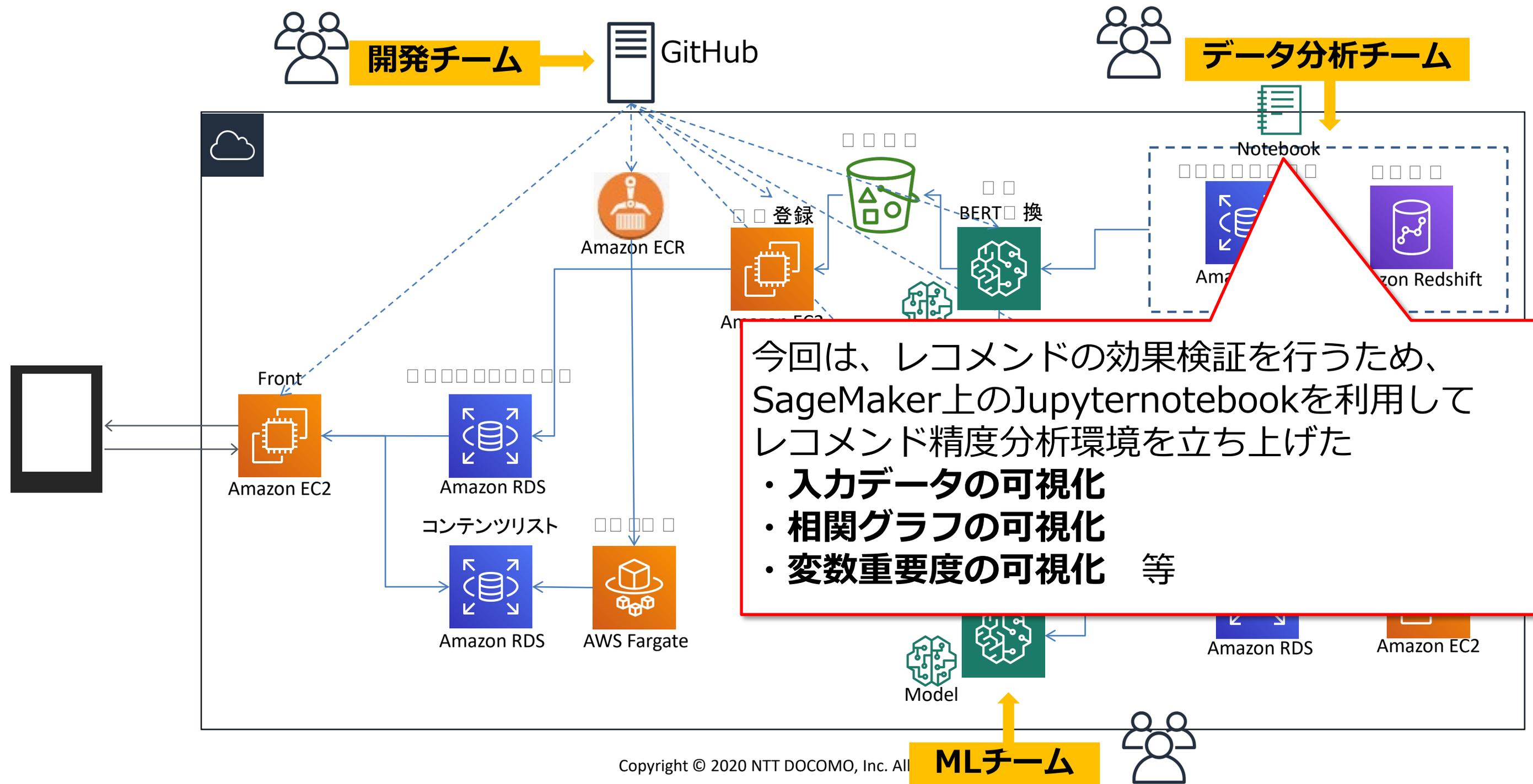


- スケーリング
- アーキテクチャへの組み込み
- アルゴリズムの変更
- チューニングの見直し

SageMaker導入のアーキテクチャ



SageMaker導入のアーキテクチャ



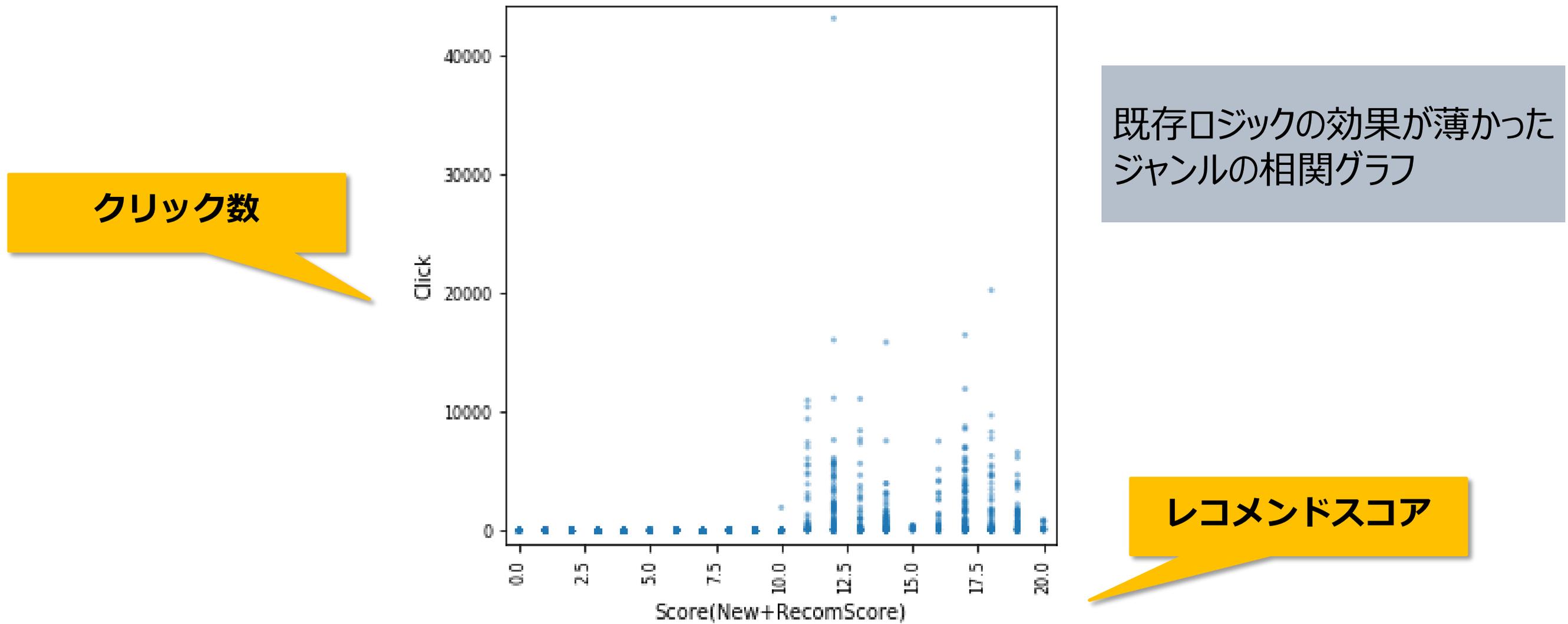
今回は、レコメンダの効果検証を行うため、SageMaker上のJupyternotebookを利用してレコメンダ精度分析環境を立ち上げた

- ・入力データの可視化
- ・相関グラフの可視化
- ・変数重要度の可視化 等

導入効果

商用導入後の改善効果

商用適用してみると一定のCTR改善効果があり！



今後はCTRやクリック数と、特徴量の関係性や寄与度などを見ながらサイクル化してモデル改善に努めていく

まとめ

SageMakerを利用してみたい

分析環境として

- **ノートブックインスタンスでjupyter環境を手軽に利用**
 - 相関分析・結果の可視化が環境を気にせず手軽に行える
- **分析リソースの確保が容易**

SageMakerを利用してみて

レコメンド機能として

- **リソースのサイジングが容易かつAWSのベースイメージが利用可能**
 - インフラ面の検討稼働削減、学習精度向上に稼働が割ける
- **ベースラインモデルの構築からハイパラチューニングがスムーズ**
 - できればS3を経由しないようにしたい
- **様々な組み込みアルゴリズムをお試し実行することができる**
 - トライ&エラーしやすい学習環境
- **学習時のみインスタンスを立ち上げておくのでコスト削減につながる**
 - 年間約20%削減、スポットインスタンスの利用も考えたい

SageMakerを利用して

SageMakerPythonSDKを利用

- サービスやリソースをオブジェクトとして扱え直感的に実装できる
 - APIコールみたいな考え方は不要
- Pythonが扱えれば初学者でもMLを動かせる
 - ハイパラとかの知識はある程度必要・管理は楽

今後やってみたいこと

- **スポットインスタンスを利用**
更なるコストの効率化
- **Amazon SageMaker Autopilot + Studioの導入**
アルゴリズムの選定やベースモデルの構築を
ある程度自動化して学習モデルの構築を効率化する

ご清聴ありがとうございました