



AWS Personalizeを活用した レコメンドシステム構築事例



齋藤 勇介

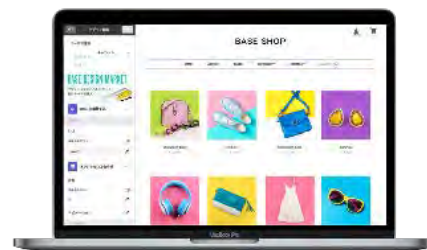
BASE株式会社
DataStrategy Team

- データ分析、機械学習を活用した開発
 - データ集計から予測モデル作成、データ配信まで何でも
 - 過去にBASEで取り組んだ事例
 - 特集コンテンツの自動生成とレコメンドサービスの取り組み
 - 機械学習にアノテーションを活用して、商品検索の関連キーワード機能を作る
- BASE開発チームブログ (<https://devblog.thebase.in/>)

1. 「BASE」 紹介
2. 「BASE」 のレコメンドシステム紹介
3. Amazon Personalizeの導入
4. 効果検証
5. 運用上のTips
6. まとめ

「BASE」 紹介

ネットショップ作成サービス「BASE」



出店ショップ数

100万ショップ以上 (個人・法人・行政を含む)

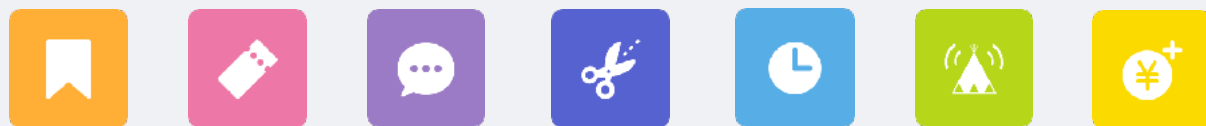
コンセプト 「誰でも簡単に使えるネットショップ作成サービス」

初期費用・月額費用 **0円**

BASEかんたん決済利用料 **3.6%+40円**

サービス利用料 **3%**

ショップオーナーのサポート機能が充実！ (BASE Apps)



個人でも決済機能をかたん導入。
審査もスピーディー！

クレジットカード



キャリア決済



銀行振込



コンビニ決済・Pay-easy

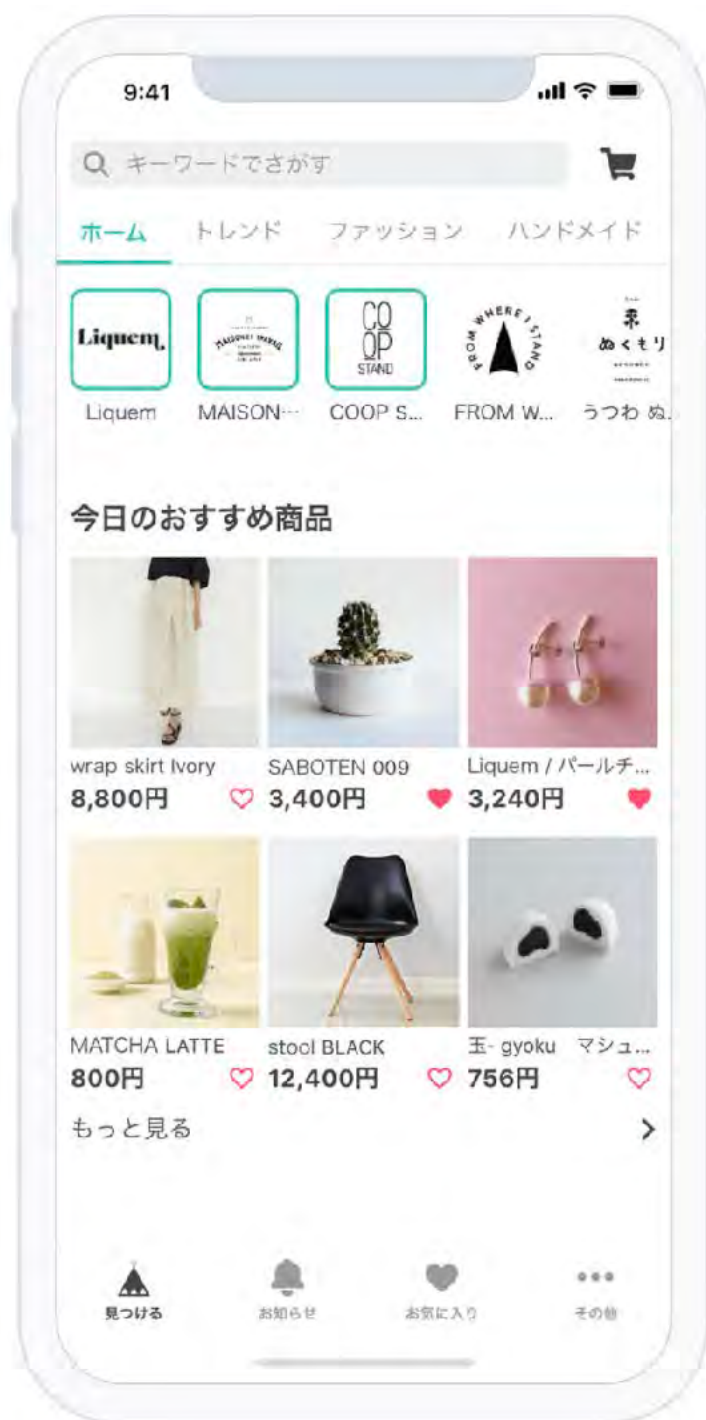


後払い



「BASE」のレコメンドシステム紹介

ショッピングアプリ「BASE」

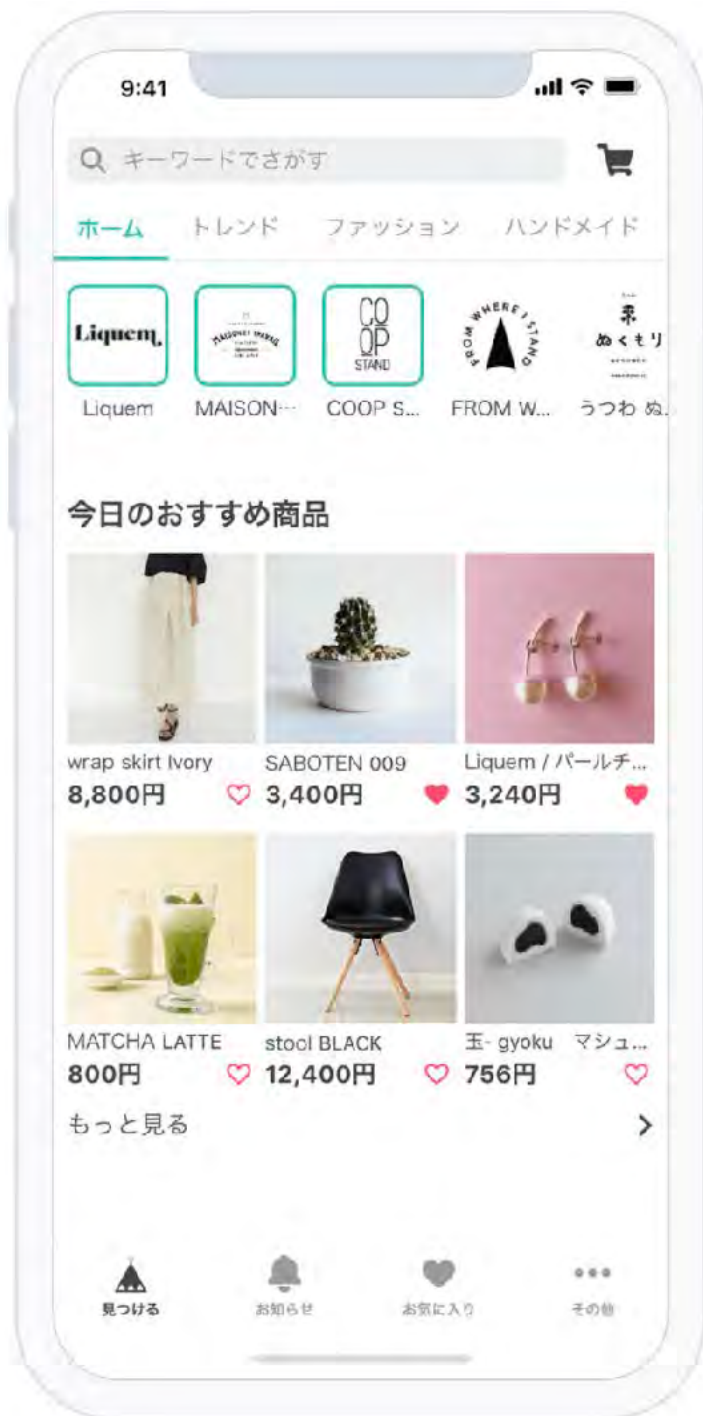


ユーザーの嗜好に合うコンテンツをレコメンド
例えばアプリのホーム画面では、

- 今日のおすすめ商品
- 今日のおすすめショップ
- テーマ別特集

にて稼働中

ショッピングアプリ「BASE」



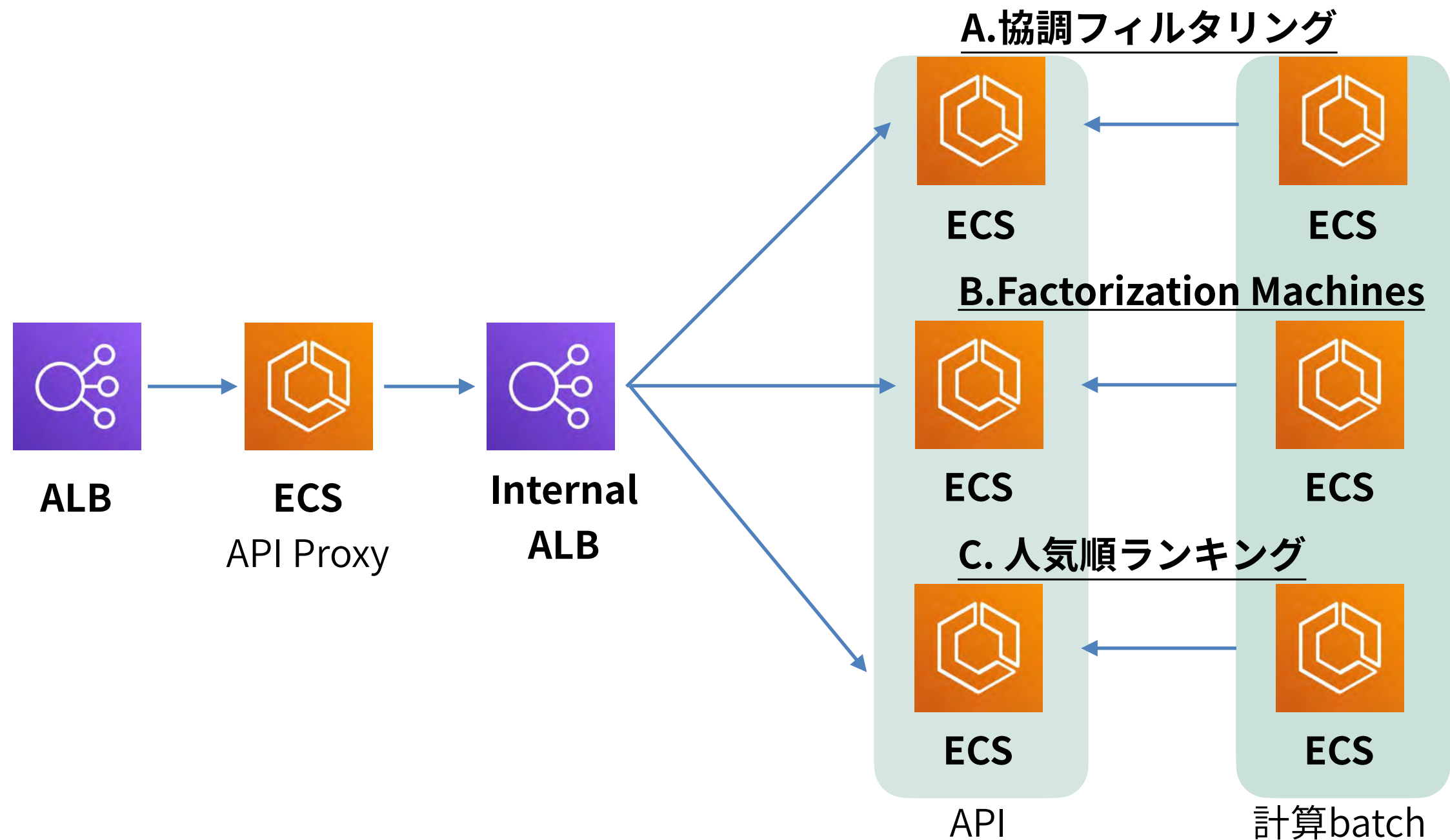
ユーザーの嗜好に合うコンテンツをレコメンド
例えばアプリのホーム画面では、

今日の話はこのエリア

- 今日のおすすめ商品
- 今日のおすすめショップ
- テーマ別特集

にて稼働中

レコメンド配信基盤



- **アルゴリズム**

- 協調フィルタリング
- Factorization Machines
- 人気順ランキング

- **予測・計算**

- リアルタイムでの予測はしておらず、バッチ計算結果を表示

- **配信ルール**

- 前段のAPI Proxyで制御

タイプ	メインアルゴリズム (予測可能ユーザー)	サブアルゴリズム (予測不可ユーザー)
A	協調フィルタリング	人気順ランキング
B	Factorization Machines	人気順ランキング

- 2種類のアルゴリズムを並行運用
- 学習に含まれていないor新規ユーザーには、各属性に合った人気商品を提示
- 複数のアルゴリズムを運用することで、推薦結果によるバイアスを受けた行動ログで学習し続けてしまう事象を軽減

一応の運用は回っていたが、
いくつか懸念点が、、

1 アルゴリズムの特性

2 メンテナンス性

3 モデル精度

- **課題**

協調フィルタリング、Factorization Machinesともに、商品全体に対するユーザーの嗜好性を推定するもので、モデル学習時に時系列性を取り入れるのが難しい
→ 直前までの行動を考慮したレコメンドが出来ない

- **どうする？**

- ユーザーのセッションベースのレコメンドモデルを作る？
 - 学習にGPU必須。環境作りなども少し億劫
- 新規モデルの開発
 - その改善のリターンは見込める？

- ・ **課題**

所々で独自実装が入っていて、担当者がいなくなった時にメンテナンスや緊急対応が難しそう

- ・ **どうする？**

- ML Opsあるある。どうする？

- **課題**

- 既存の運用モデルはベストな選択なのか？
- より良いモデルは存在するのでは？
- 学習パラメーターもある程度見直す必要があるが、放置状態

- **どうする？**

- 新規モデルの開発
 - その改善のリターンは見込める？
- 学習パラメーターの検証
 - やりたいけどやっていない

Amazon Personalizeの導入

2019年6月 Amazon Personalizeが 東京リージョンで利用可能に

2019年10月 「BASE」 で Amazon Personalizeの運用開始

- ・ **アルゴリズムの特性**

「HRNN」という定義済みアルゴリズムを使えば、セッションベースの学習が可能

- ・ **メンテナンス性**

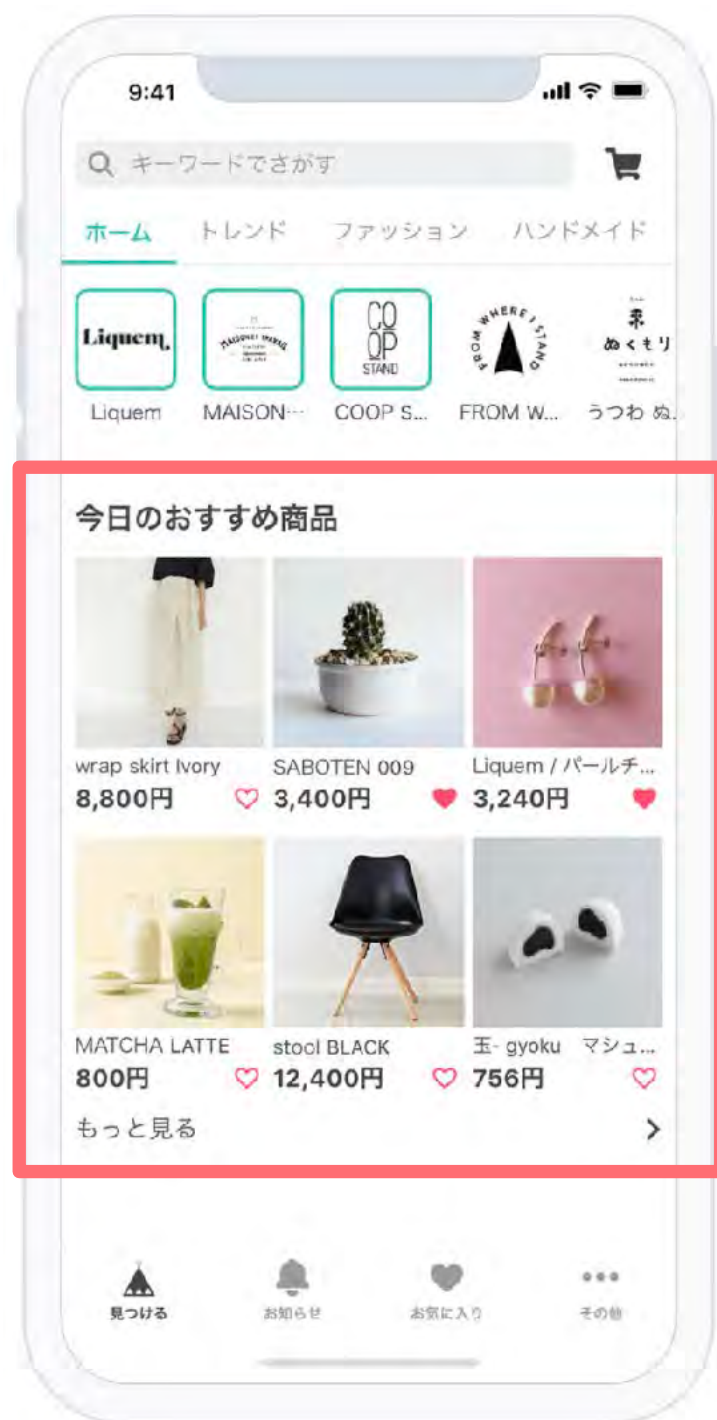
最低限、csvを3file用意すれば大丈夫
運用開始から8ヶ月経過するも、コード修正は2回だけ
障害も発生せず

- ・ **モデル精度**

最適パラメーターの探索も設定次第で簡単に可能

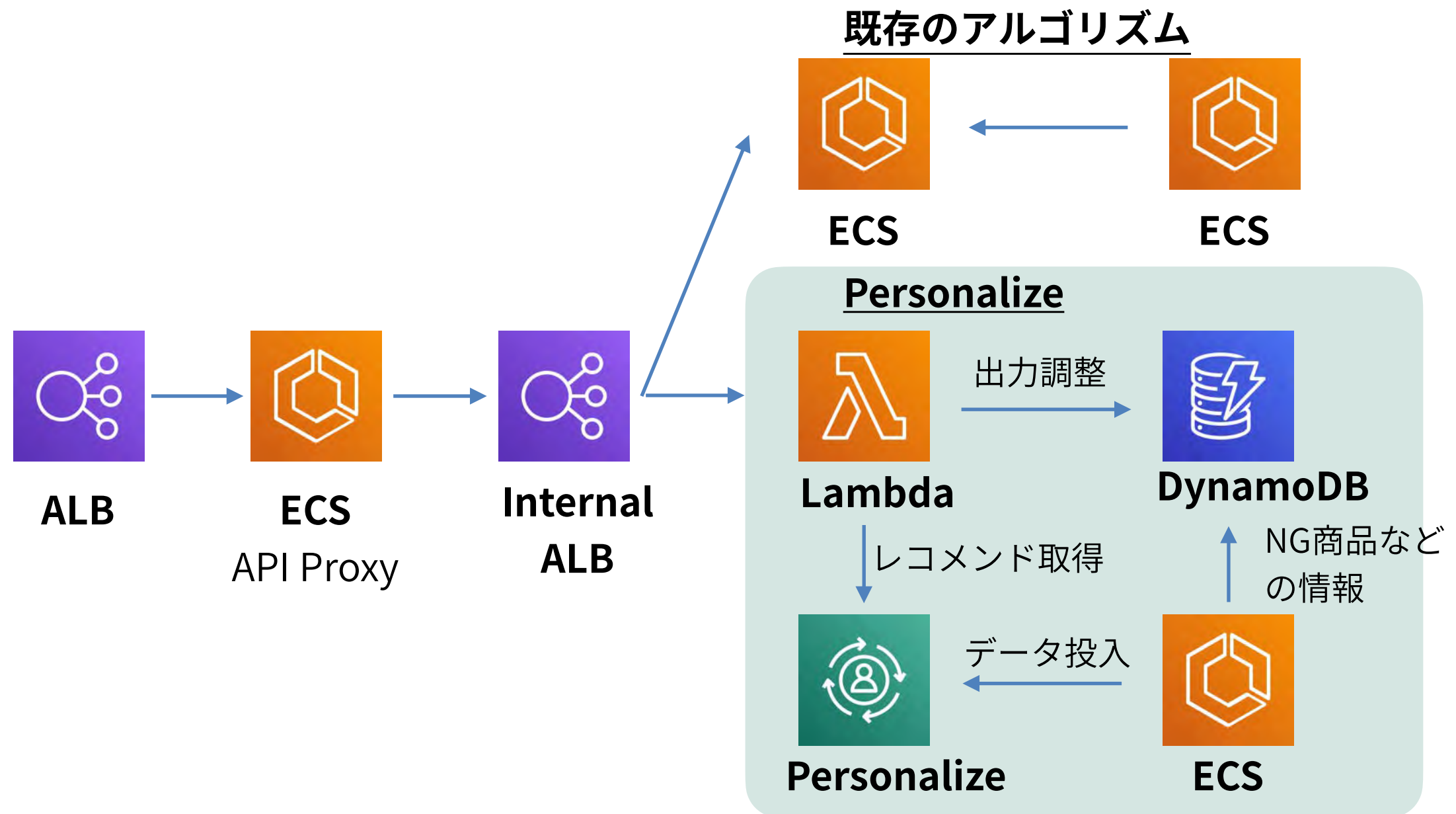
**色々と理由を挙げましたが、
どれぐらいの精度が出るものか検証してみたい
という動機が一番大きかったです**

ショッピングアプリ「BASE」



「今日のおすすめ商品」枠
にて、検証開始

レコメンド配信基盤



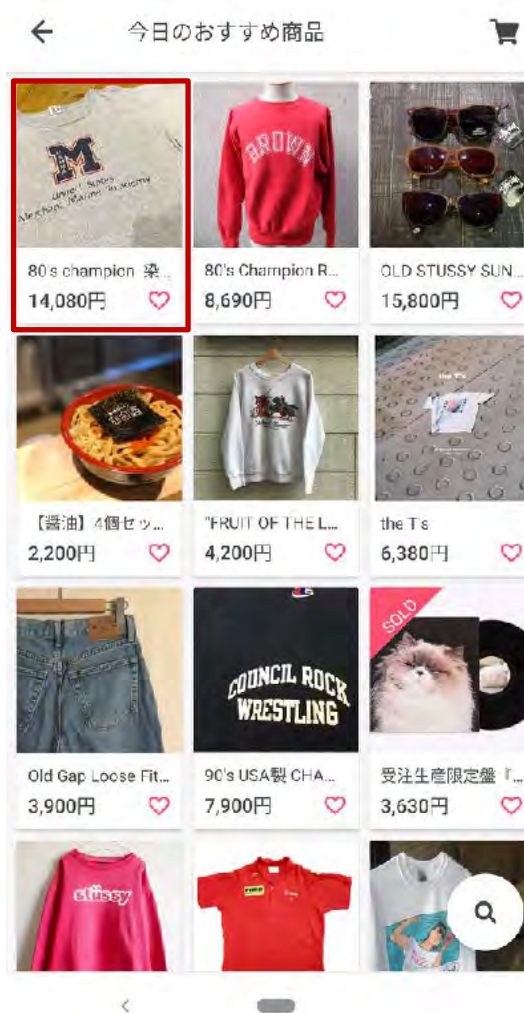
効果検証

1. 閲覧率



アプリホーム画面

2. お気に入り率

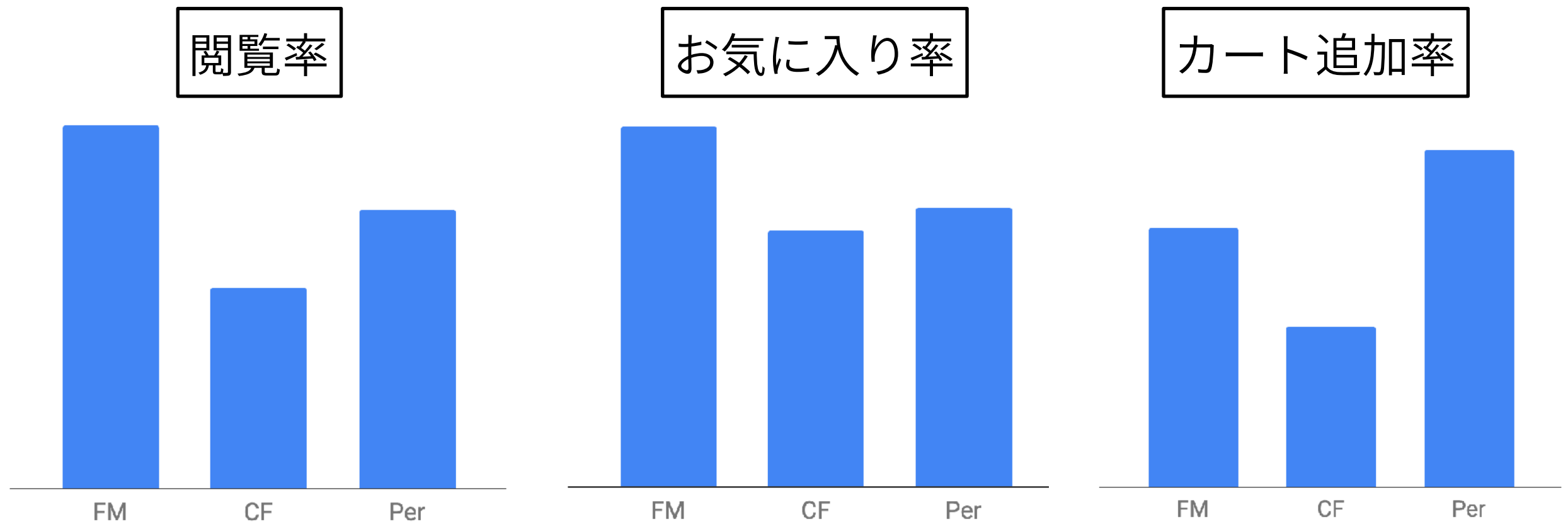


もっと見る画面

3. カート追加率



商品詳細画面



- Personalize(Per)は、カート追加率で最も優れた予測を実現
- 閲覧率では、既存のモデルに勝てないこともある
 - アルゴリズムの特性、カスタマイズ性に起因
- 回遊性を重視するのか、売上を重視するのかでより良い選択を

運用上のTips

最重要！

必要な時だけ、ハイパーパラメーターチューニングする

Perform HPO / Perform AutoMLというパラメーターの仕様をよく理解して、毎回チューニングしないようにする

ちなみに、
運用中のデータサイズでチューニングすると、
チューニングなし学習の約15回分の請求がきます

学習時に含まれていないユーザーに人気順のitemが出る

新規ユーザーや長期非ログインユーザーに対して、
共通のレコメンドがされ、
Personalize内での調整が今の所できない

別途取得したユーザー属性元にレコメンドしたい場合は、
追加で拡張する必要がある

まとめ

Amazon Personalizeを使ってみて



- 手間を掛けずにレコメンドサービスを開発できるツール
- メンテナンスコストも削減可能
- ハイパーパラメータチューニングも容易

ご清聴ありがとうございました