



AI を用いたタクシー配車における BigQuery 徹底活用術

老木 智章

株式会社 Mobility Technologies AI 技術開発部
データサイエンティスト

タクシーアプリ「GO」と BigQuery

タクシーアプリ「GO」では、運用ログを用いた機能として、
到着時間予測、待ち時間予測、AI 予約などを提供している。

各機能に必要な、

- 数万台のタクシー位置情報の複雑な集計処理
- AI の学習・推論処理

を BigQuery 上で完結させることで、

エンジニアリング コストの削減とリアルタイム処理を実現した。

計算処理基盤としての BigQuery の活用について紹介する。



アジェンダ

1. 提供サービス例
2. BigQuery による集計処理 ～複雑な処理の実装～
3. BigQuery による集計処理 ～リアルタイム性～
4. BigQuery ML での予測

提供サービス例

01

サービス例：AI 予約と従来の予約

従来：車両を事前確保する。乗務員の **拘束時間が長い**。

AI 予約：予約時間直前に車両を確保する。 **拘束時間が短い**。

AI 予約では予約時間に周辺に車両が存在する保証がない。

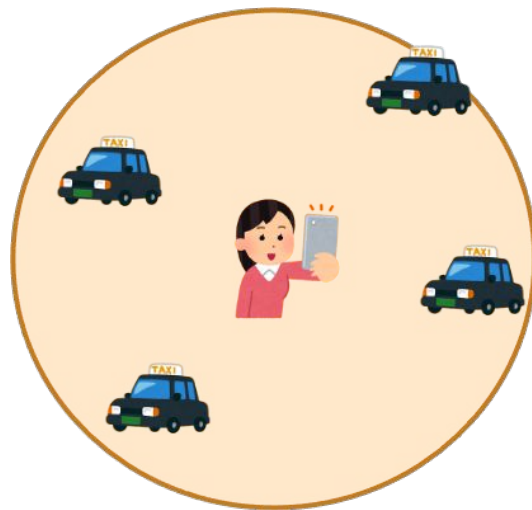
そこで、予約時間に車両が手配できる確率を求め、

確率が高い時のみ、予約を受け付ける。

確率算出のためには、予約された時間・予約された場所に

どの程度車両がいるか（車両密度）の情報などが必要。

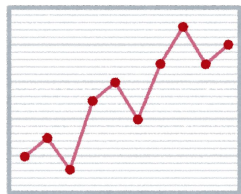
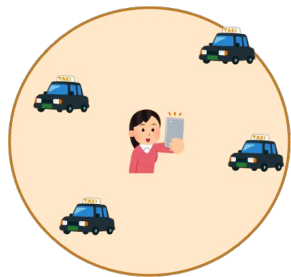
12:30 に予約された場合、
12:15 頃から周辺車両を手配する



位置情報からサービス提供までの流れ

データサイエンティストが開発する 1, 2 を BigQuery で完結させることで、開発・保守負荷を低減

1. 位置情報を集計



2. 集計値からモデルを学習

3. モデルの予測結果からサービス提供



BigQuery による集計処理

- 複雑な処理の実装 -

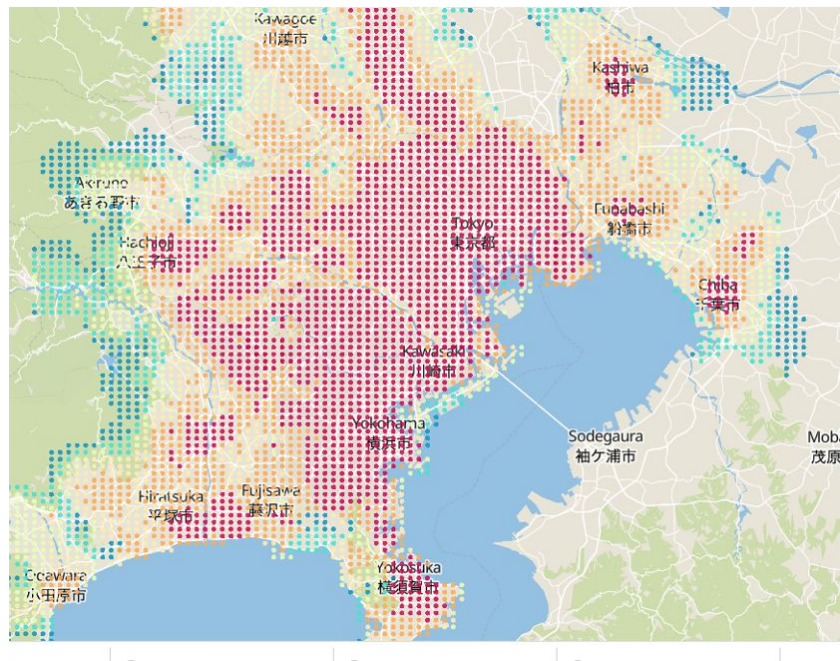
02

位置情報の集計処理

15 分ごとに、どこに何台の **配車可能車両** がいたかを集計している。位置情報は数秒周期で数万台分あるため、かなりの計算能力が必要。

この集計処理を BigQuery で行うことで、**並列処理を簡単に実装できる**。一方、

- 複雑な処理の SQL での実装
- 15 分に処理をおさめるエンジニアリングが必要となる。



kepler.gl による車両カウント可視化例

BigQuery GIS による集計

BigQuery GIS を用いることで、地理情報の処理を簡単に SQL で記載できる。(しかも速い！)

右のクエリでは、メッシュごとに異なる配車距離半径に入る車両の台数をカウントしている。

```
SELECT
  m.mesh_code,
  COUNT(*)
FROM mesh_points AS m, cars
WHERE
  ST_DISTANCE(
    ST_GEOGPOINT (m.lon, m.lat),
    cars.point
  ) < m.dispatch_limit
GROUP BY mesh_code
```

複雑な処理の SQL での実装

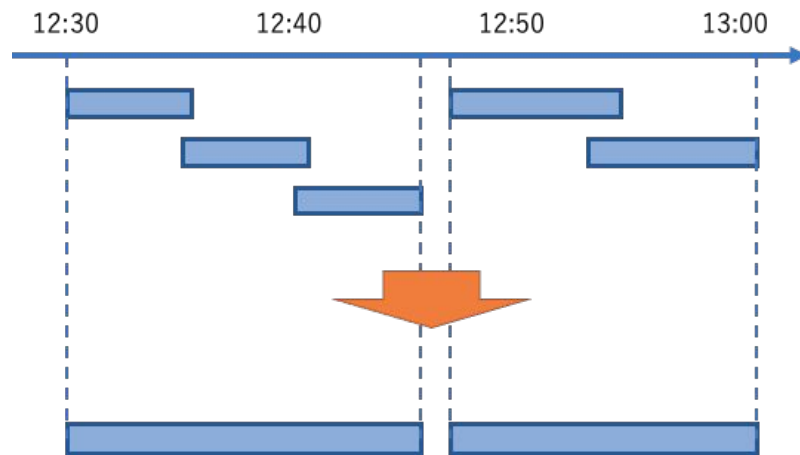
位置情報の集計では、複雑な業務ルールを SQL で実現している。

複雑な処理を実現するための機能として、

- javascript UDF
- 分析関数
- ARRAY / STRUCT
- ASSERT

を用いている。

これらでも記述が困難なケースでは、
jinja2 を用いて SQL を生成している。



重複する時間帯のマージ処理も
分析関数を使えば記述できる

jinja2 による SQL の生成

jinja2 はループを使って、SQL を生成できる。

右のクエリは 2 の 10 乗を求めるサンプル。

二分探索のように、一定のループ回数で収束が期待できる処理を記述できる。

このようにさまざまな手段を用いることで、SQL 上で多様な処理を実現できた。

```
WITH base0 AS (  
    SELECT  
        2 as v  
)  
  
{% for n in range(1, 10)%}  
    , base{{n}} AS (  
        SELECT  
            v * 2 as v  
        FROM base{{n-1}}  
    )  
{% endfor %}  
  
SELECT  
    v  
FROM base9
```

BigQuery による集計処理

- リアルタイム性 -

03

リアルタイム性と料金プラン

我々の処理は 15 分以内に処理する必要がある。

BigQuery の処理速度は料金プランが影響する。

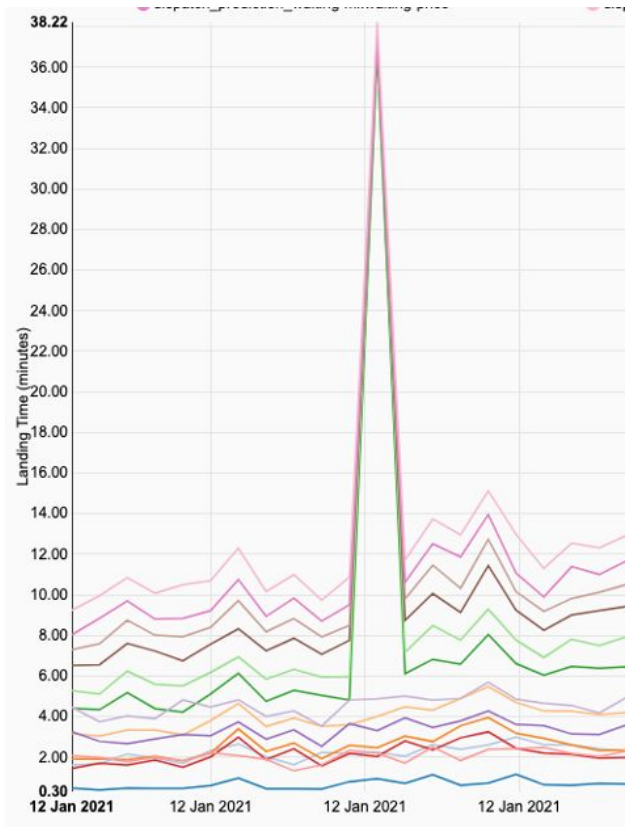
オンデマンドプラン

スロット割当数に上限があり、割当量の **保証なし**。

定額プラン

スロット割当量への課金のため割当 **保証あり**。

オンデマンド利用するので、スロット割当内で処理するための **高速化**と、処理時間の **不安定性への対処**が必要。

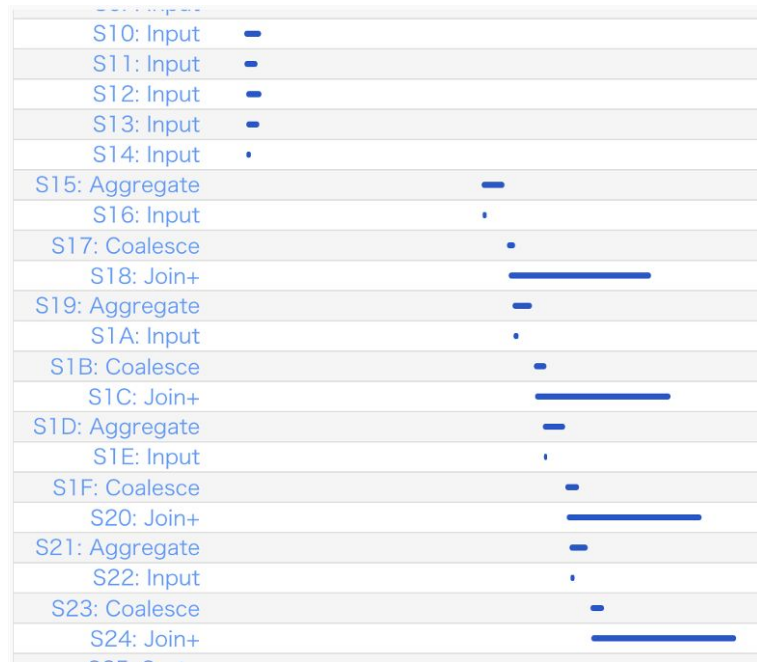


クエリ実行時間がたまに長くなる例
縦軸が実行時間

オンデマンドでのサービス提供

クエリ的高速化には bq-visualizer を用いてボトルネックとなる処理を特定し、SQL の改良を行っている。

不安定性の対処には、処理が遅延した際に、過去に計算した統計値へフォールバックさせている。



bq-visualizer の出力例

BigQuery のおもしろ高速化例

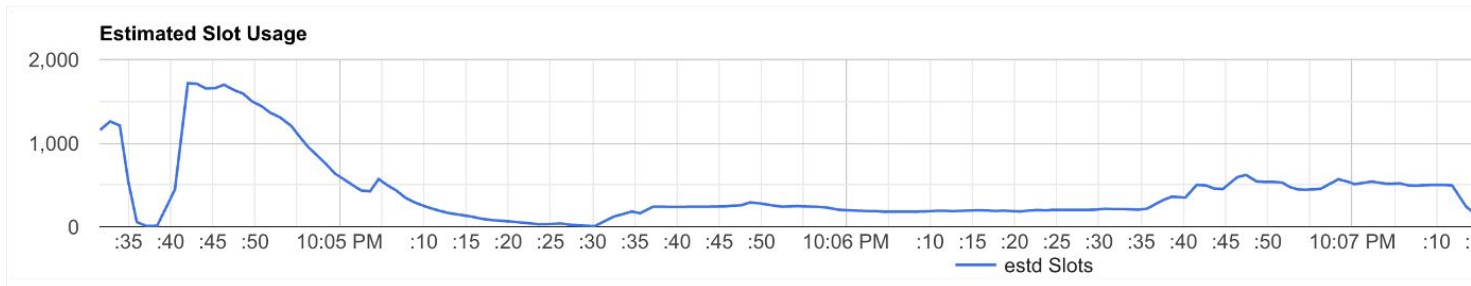
BigQuery は他の DB 製品とアーキテクチャが異なるため、高速化テクニックも異なる。

特異な例として、並列処理エンジンという特性から、並列度を上げるテクニックがある。

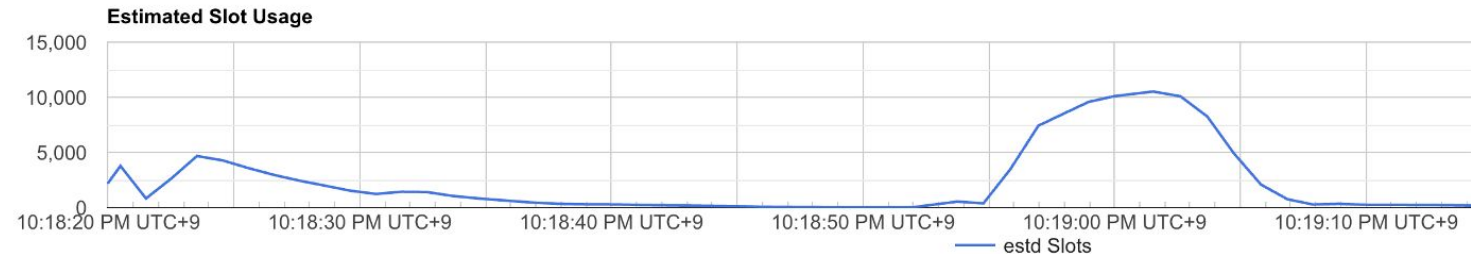
```
WITH BASE AS (...),  
MULTIPLE_READ as (  
  SELECT * from BASE WHERE MOD(user_id, 3) = 0  
  UNION ALL  
  SELECT * FROM BASE WHERE MOD(user_id, 3) = 1  
  UNION ALL  
  SELECT * FROM BASE WHERE MOD(user_id, 3) = 2  
)  
SELECT  
  重い処理  
FROM MULTIPLE_READ
```

並列度向上効果例

最適化前



最適化後



集計処理まとめ

- 処理を SQL に集約することで、サービスの実装・保守コストを低減した。
- BigQuery の並列処理能力により、簡単に大規模なデータを処理できた。
- SQL 上で処理を行うため、BigQuery の各種機能を活用した。



AI を用いたタクシー配車における BigQuery 徹底活用術

島越 直人

株式会社 Mobility Technologies AI 技術開発部
データサイエンティスト

BigQuery ML による予測

04

BigQuery ML について

- BigQuery 上で機械学習モデルの訓練・予測が行える
 - 線形回帰などの基本的なものから XGBoost やMLP、AutoML にも対応
- GCS などに推論データなどを外出しする必要がなく、**推論速度の向上**が期待できる
- 推論用の GKE pod などを立てる必要がなく、**管理コストが低い**

```
CREATE MODEL `mydataset.linear_model`\nOPTIONS(\n  MODEL_TYPE="LINEAR_REG",\n) AS\n(\n  SELECT\n    column1,\n    column2,\n    column3,\n    label\n  FROM\n    mydataset.train_dataset\n)
```

GO の AI 予約における活用事例



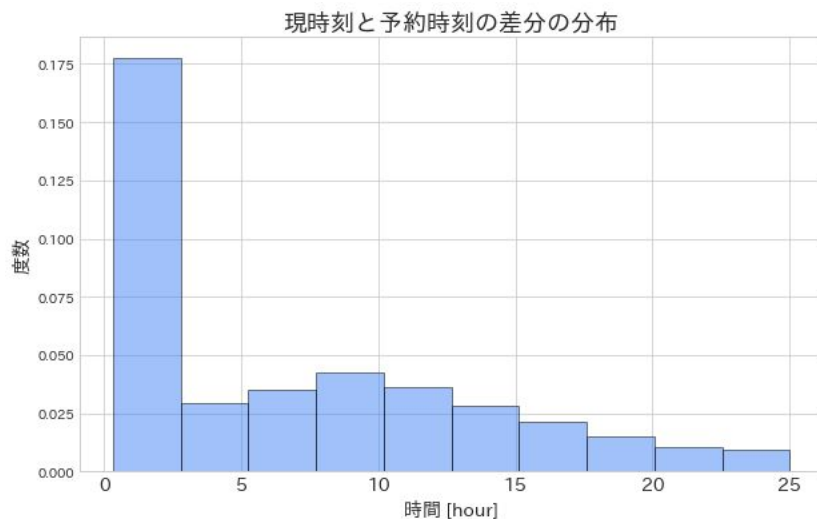
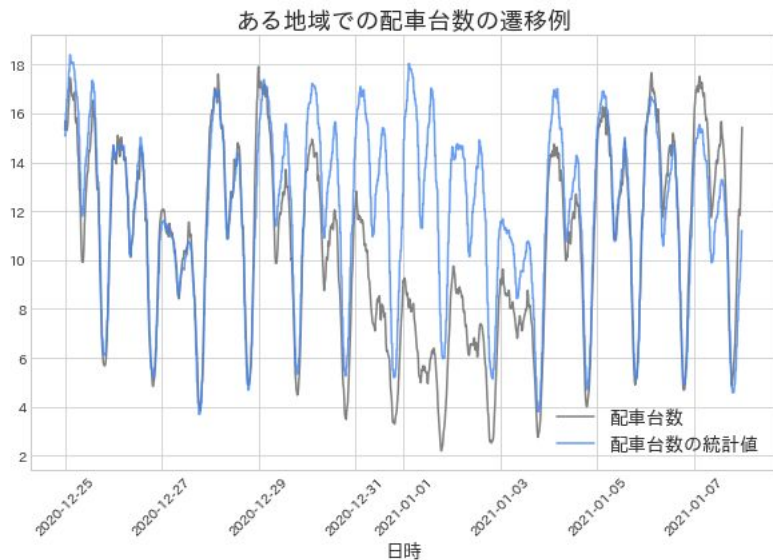
25 分後から 7 日先までの
予約が可能

事前に予約時間前後の
配車可能数の予測が必要

何故予測が必要なのか

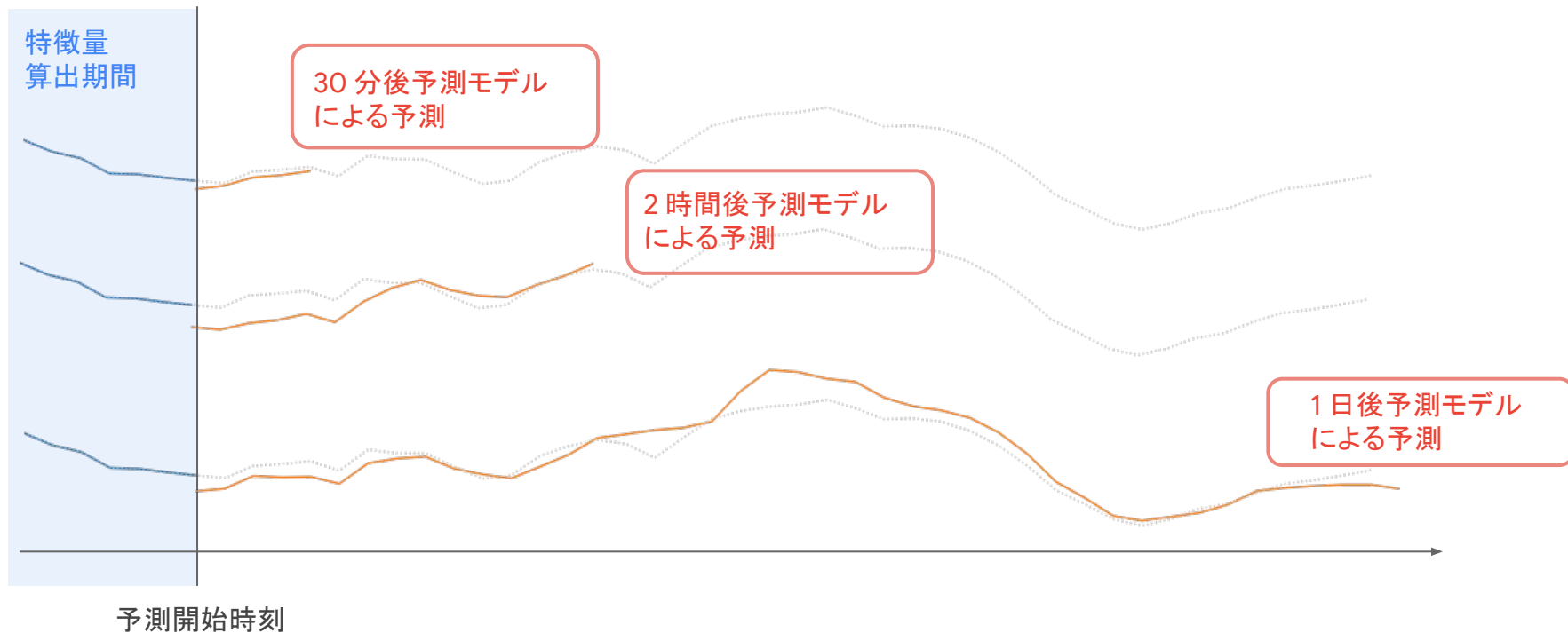
AI 予約における課題

1. 祝日や年末年始など、単なる統計値では実測値と大きく外れる場合がある
2. 直前に予約を行うユーザーが多く、リアルタイム性が重要



GO の AI 予約における予測モデル

何分後を予測するかによって複数モデルを作成



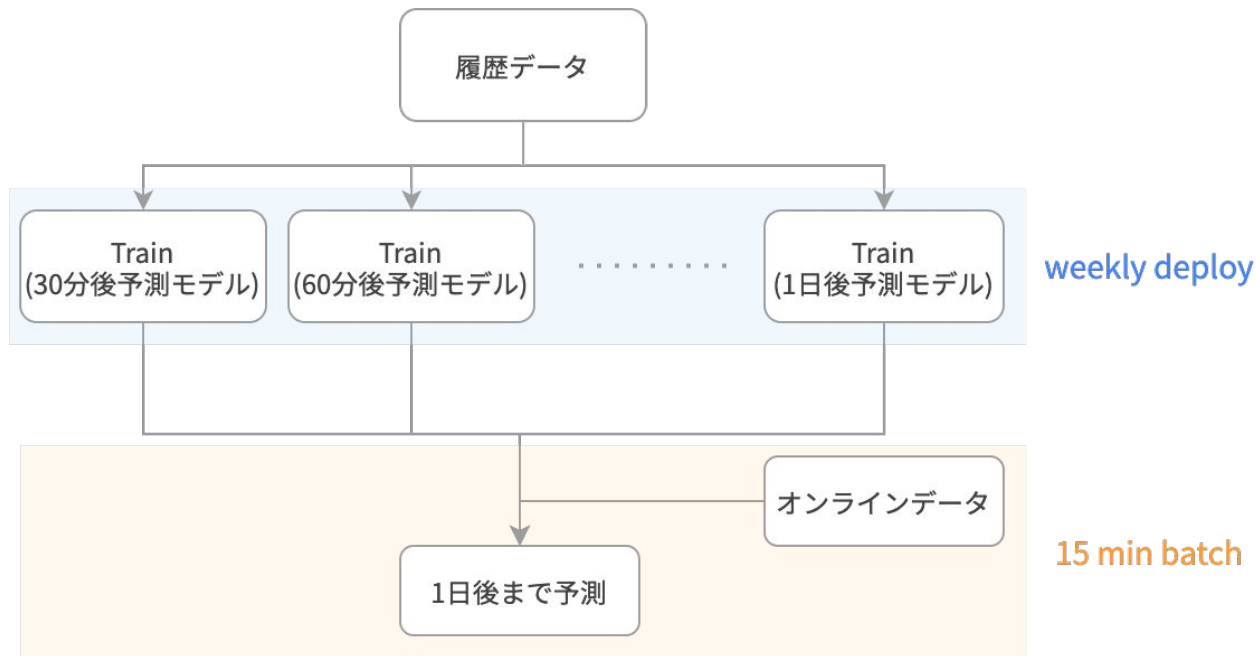
Jinja2 テンプレートを使用したクエリ例

N 分後予測モデル毎に BigQuery ML モデルを作成

```
CREATE OR REPLACE MODEL `{{dataset_id}}.model_after_{{minute}}min`
OPTIONS(
  MODEL_TYPE="BOOSTED_TREE_REGRESSOR",
  DATA_SPLIT_METHOD="CUSTOM",
  DATA_SPLIT_COL="is_eval",
  INPUT_LABEL_COLS=['{{label_name}}']
) AS
(
  SELECT
    ts_bin > TIMESTAMP_ADD("{{start_ts}}", INTERVAL {{train_n_days}} DAY) AS is_eval,
    {{label_name}},
    AVG({{label_name}}) OVER (
      PARTITION BY mesh_code
      ORDER BY UNIX_SECONDS(ts_bin)
      RANGE BETWEEN {{stats_sec}} PRECEDING AND {{lag_sec}} PRECEDING
    ) AS avg_{{label_name}}
  FROM
    dataset
)
```

N 分後予測モデル毎に
特徴量算出期間を変化させる

BigQuery ML を用いた予測パイプライン



予測も BigQuery のみで**全て完結！**

BigQuery ML の実用上の注意点 ①

線形モデルなどを用いる場合は、スキャン量を減らすために中間テーブルを作成する
(定額料金プランの場合は定額料金の範囲内に収まる)

操作	料金 (US リージョン)
線形回帰モデルの訓練	\$ 250.00 / TB 通常の 50 倍 !!
ロジスティック回帰モデルの訓練	
K 平均法クラスタリングモデルの訓練	
時系列モデルの訓練	
AUTO ML	\$ 5.00 / TB + AI Platform Training 費用
DNN	
ブーストツリーモデル	
予測	\$ 5.00 / TB

BigQuery ML の実用上の注意点 ②

ブースト ツリーモデルや DNN を使用する場合、モデルの更新時に注意

- Train ステージに入って暫くした段階で model の置き換えが行われてしまう
- 一時的なモデル名で訓練させ、訓練終了後に元のモデル名にコピーするといった運用が必要

ステージ ?

✓ Preprocess 15.353 秒

🔄 Train



sample_model

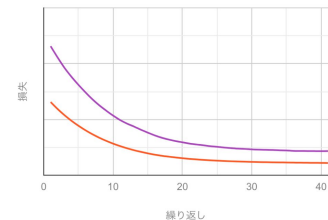
詳細 トレーニング 評価 スキーマ

表示形式

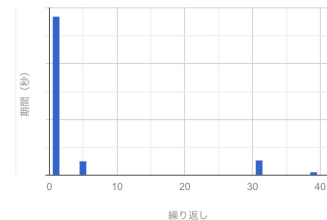
☒ グラフ

☐ 表

損失



期間 (秒)



BigQuery ML の課題

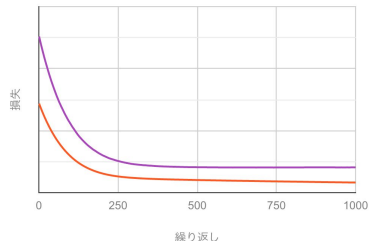
EARLY STOPPING の挙動

~~「1 epoch でもMIN_REL_PROGRESS 分だけ改善しなければ停止」~~

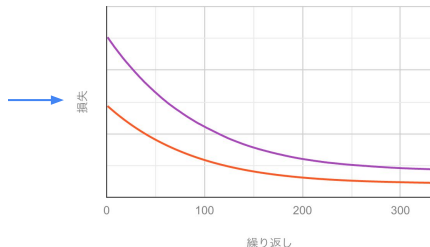
「N epoch の間改善しなければ停止」

まだまだ改善が見込める場合も学習が止まる

損失



損失



MAX_ITERATIONS の最大値が 1000 に制限

(特に) XGBoost を用いる場合は木の数が 1000 以上になることは往々にしてあり得る

1000 iter 以内に損失関数が下がりきるように
学習率をチューニング する必要がある

手元で学習させた方が
現状だと精度が高くなることが多い

BigQuery ML まとめ

- タクシー配車アプリ「GO」では、BigQuery ML を活用することでサービスに必要な予測機能をコンパクトに実現した。
- AI 予約を例に BigQuery ML を用いて予測パイプラインを構築した例を紹介した。
- BigQuery ML を実務で用いる際の注意点や課題について紹介した。

Thank you.

