



# Google を支える 機械学習サービス監視の MLOps

佐藤 一憲

Google デベロッパー アドボケイト

# スピーカー自己紹介



佐藤一憲

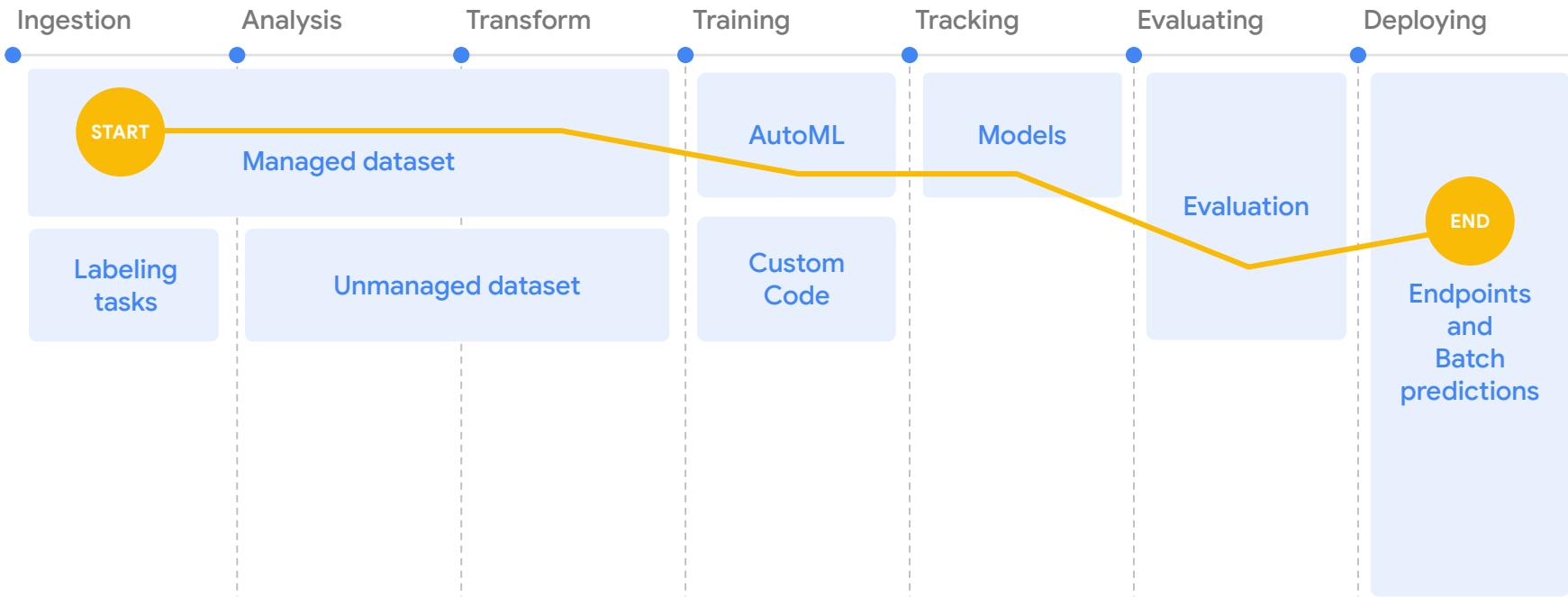
Google  
デベロッパー アドボケイト

Google Cloud のデベロッパー アドボケイトとして、  
機械学習や AI 系プロダクトの開発者支援を担当。  
Google Cloud Next、Google I/O、NVIDIA GTC 等の  
主要イベントでスピーカーを務め、Google Cloud 公式ブログに  
多数の記事を寄稿。また Google Cloud 開発者コミュニティを  
10 年以上にわたり支援している。



# Google の MLOps

# ML 開発のライフサイクル

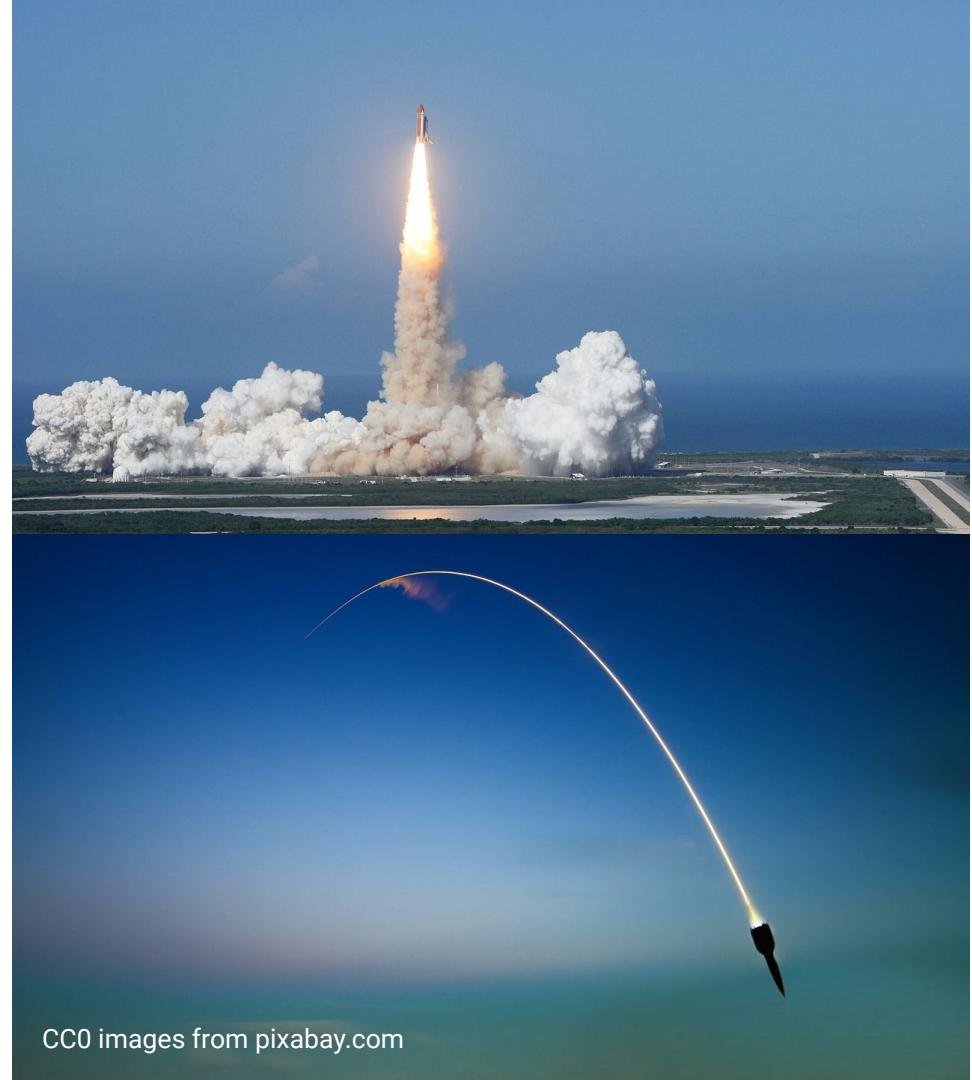


# MLOps

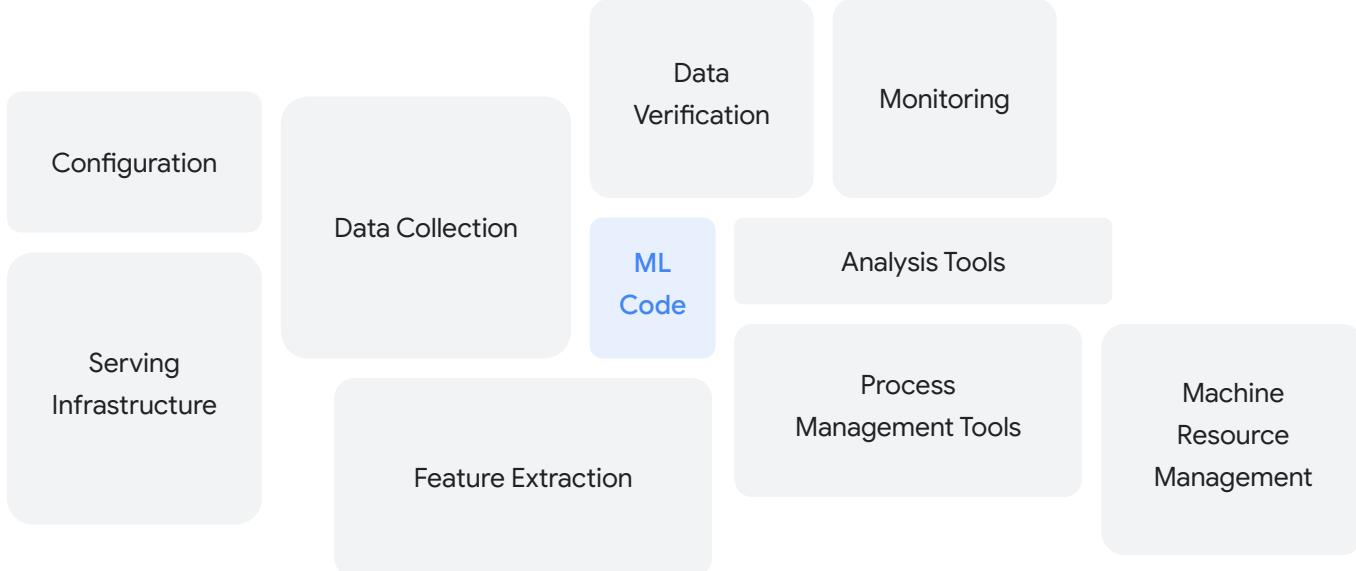
---

ML システムの開発(Dev)と運用(Ops)を統合  
する開発文化とプラクティス

ローンチは簡単、  
運用が難しい。



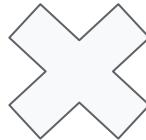
CC0 images from pixabay.com



# ML システム実運用の課題

## ML 固有の課題

- データ、特徴量、モデル、パイプライン、実験結果等の管理
- 繼続学習とデプロイ、監視
- スキューとドリフト
- 学習データの検証
- 学習後モデルの分析
- 公平性と説明可能性



## 実運用環境の課題

- スケーラビリティ
- 可用性
- ポータビリティ
- 再現可能性
- モジュール性
- 監視とアラート
- セキュリティ
- サーバレス

# Google における ML 実運用環境の一例

Google 最大規模のある ML サービスの運用体制:

- 15 年以上の運用実績
- ML SREs による運用
- 数千種類の ML モデルを一時間おきに継続学習
- グローバルのサービス インフラに継続デプロイ

# Google における ML 実運用環境の歴史

Continuous Training for Production ML in the TFX Platform. OpML (2019).

Slice Finder: Automated Data Slicing for Model Validation. ICDE (2019).

Data Validation for Machine Learning. SysML (2019).

TFX: A TensorFlow-Based Production-Scale Machine Learning Platform. KDD (2017).

Data Management Challenges in Production Machine Learning. SIGMOD (2017).

Rules of Machine Learning: Best Practices for ML Engineering. Google AI Web (2017).

---

---

## The High-Int

---

---

D. Scull  
Todd Phillips  
{toddphilli

### ABSTRACT

Creating and maintaining a platform for reliable and deploying machine learning models require orchestration of many components – a need for model versioning, data management, for isolating both data as well as models, and finally for serving models in production. This is becoming challenging when data changes over time and I need to make sure my models are up-to-date. Underpinning this is a need for automation. Underpinning orchestration is often done ad hoc using glue code scripts developed by individual teams for specific leading to duplicated effort and fragile code.

We present TensorFlow Extended (TFX), a general-purpose machine learning platform at Google that integrates machine learning into ML pipelines. We were able to standardize inputs, simplify the platform configuration, and time to production from the order of months to days.

A screenshot of a web page titled "Data Management". The title is at the top left. Below it is a large section with the heading "Neoklis Polyzotis, Sudipto Mukherjee" and the subtitle "(npolyzotis, smukherjee@mit.edu)". To the right of this is a large gray box containing the word "Best". Below the main title, there's a section titled "ABSTRACT" followed by a detailed paragraph. Further down, there are sections for "CCS Concepts", "Terminology Overview", and "Before Machine Learning". At the bottom, there are three numbered rules: Rule #1, Rule #2, and Rule #3.

---

# DATA VALIDATION FOR MACHINE LEARNING

---

Eric Breck<sup>1</sup> Neoklis Polyzotis<sup>1</sup> Sudip Roy<sup>1</sup> Steven Euijong Whang<sup>2</sup> Martin Zinkevich<sup>1</sup>

---

## ABSTRACT

Machine learning is a powerful tool for gleaned knowledge from massive amounts of data. While a great deal of machine learning research has focused on the correctness and efficiency of training and inference algorithms, there is less attention in the equally important problem of measuring the quality of data fed to machine learning. The importance of this problem is hard to dispute: errors in the input data can nullify any benefits on speed and accuracy for training and inference. This argument points to a data-centric approach to machine learning that treats training and serving data as an important production asset, on par with the algorithm and infrastructure used for learning.

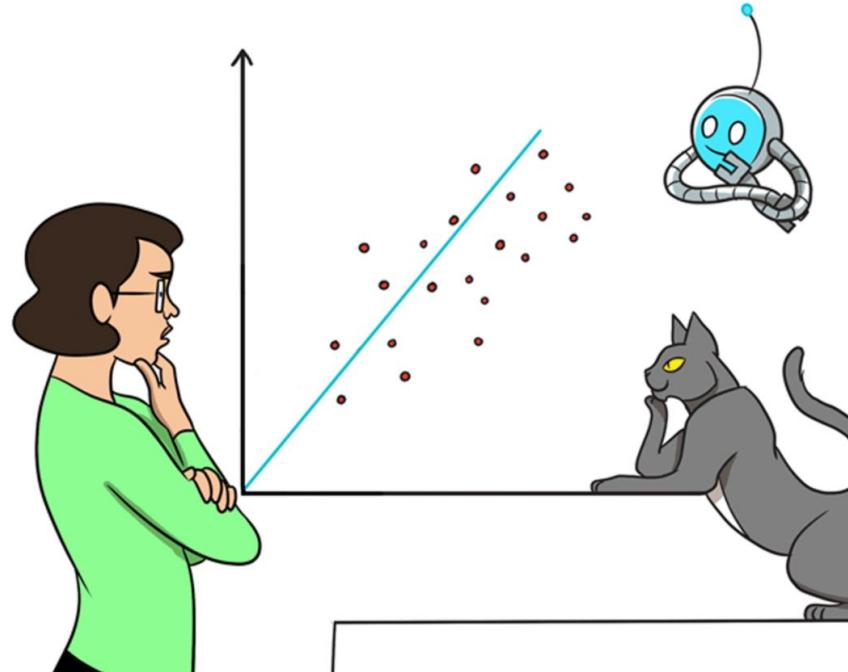
In this paper, we tackle this problem and present a data validation system that is designed to detect anomalies specifically in data fed into machine learning pipelines. This system is deployed in production as an integral part of FTX(Bayoler et al., 2017) – an end-to-end machine learning platform at Google. It is used by hundreds of production teams across Google to serve as a safety net for several hundred machine learning models today. We faced several challenges in developing our system, most notably around the design of ML pipelines to validate in the face of unexpected patterns, schema-free data, or training/serving skew. We discuss these challenges, the techniques we used to address them, and the various design choices that we made in implementing the system. Finally, we present evidence from the system's deployment in production that illustrate the tangible benefits of data validation in the context of ML: early detection of errors, model-quality wins from using better data, savings in engineering hours to debug problems, and a shift towards data-centric workflows in model development.



# 機械学習モデルが うまく動かないとき

# なぜ ML モデルが期待通りに動かないのか？

あらゆる ML モデルは、  
実世界とのズレが発生する時が来る  
ML サービスの性能劣化として現れる



Google

Google Cloud

# スキュードリフト

**Training-Serving Skew:**

トレーニング時とサービス時のズレ

**Prediction Drift:**

サービス時の経時的なズレ



Grandbrothers

Google Cloud

## Google で発生した Training-serving skew によるトラブル

- ある ML パイプラインで毎日モデル学習を実行
- あるエンジニアのコード改修により、特定の特徴量が -1 に固定
- ML モデルの推論ではエラーが発生しないため、誰も気づかない
- 推論結果はつぎの学習にも利用され、状況は静かに悪化

“

**We want the user to treat data errors  
with the same rigor and care that they  
deal with bugs in code”**



コードのバグ検出と同じ真剣さでデータのバグを見つけるべし

- [TFX paper](#)

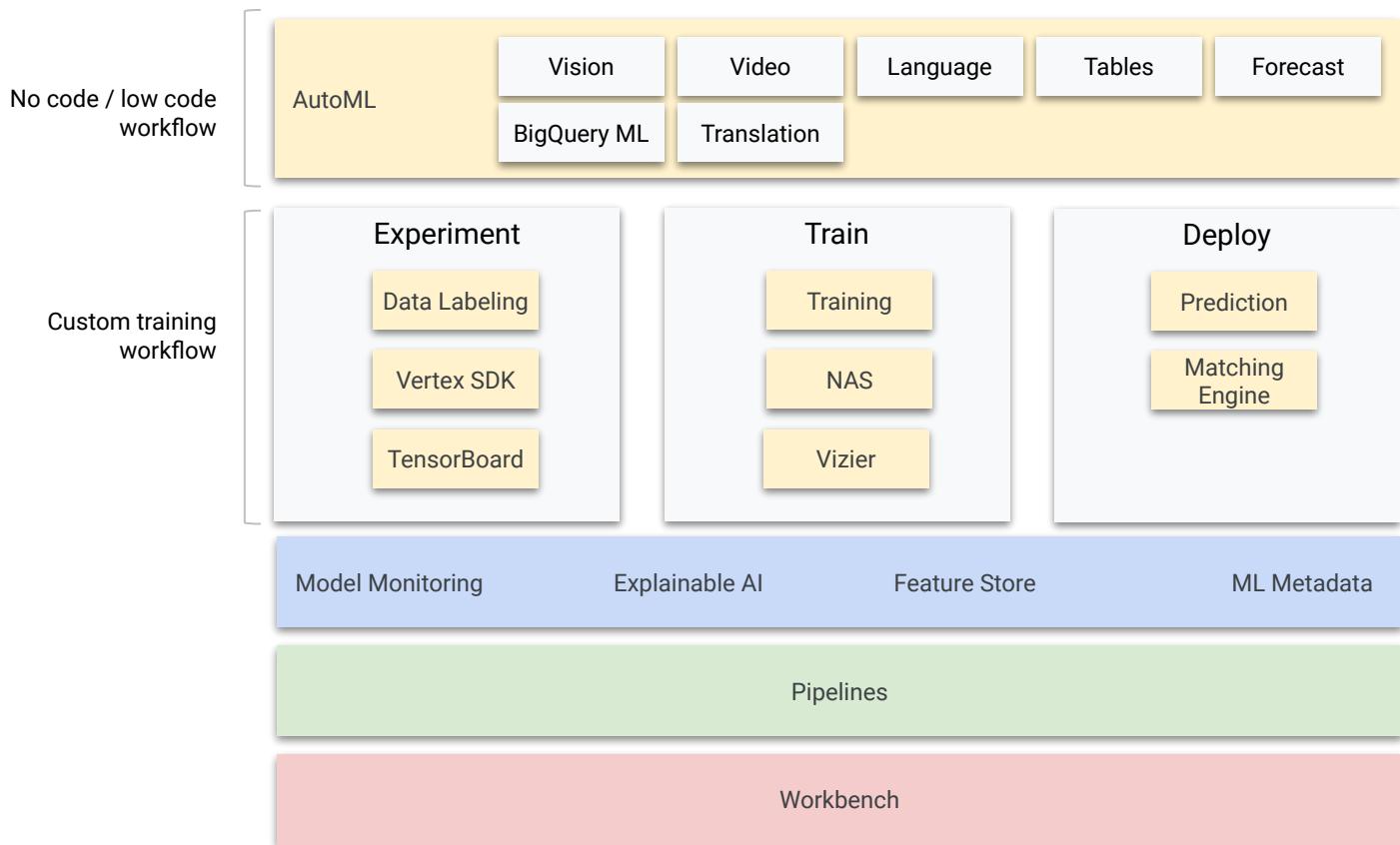


# Vertex AI Model Monitoring による特徴量分布の監視

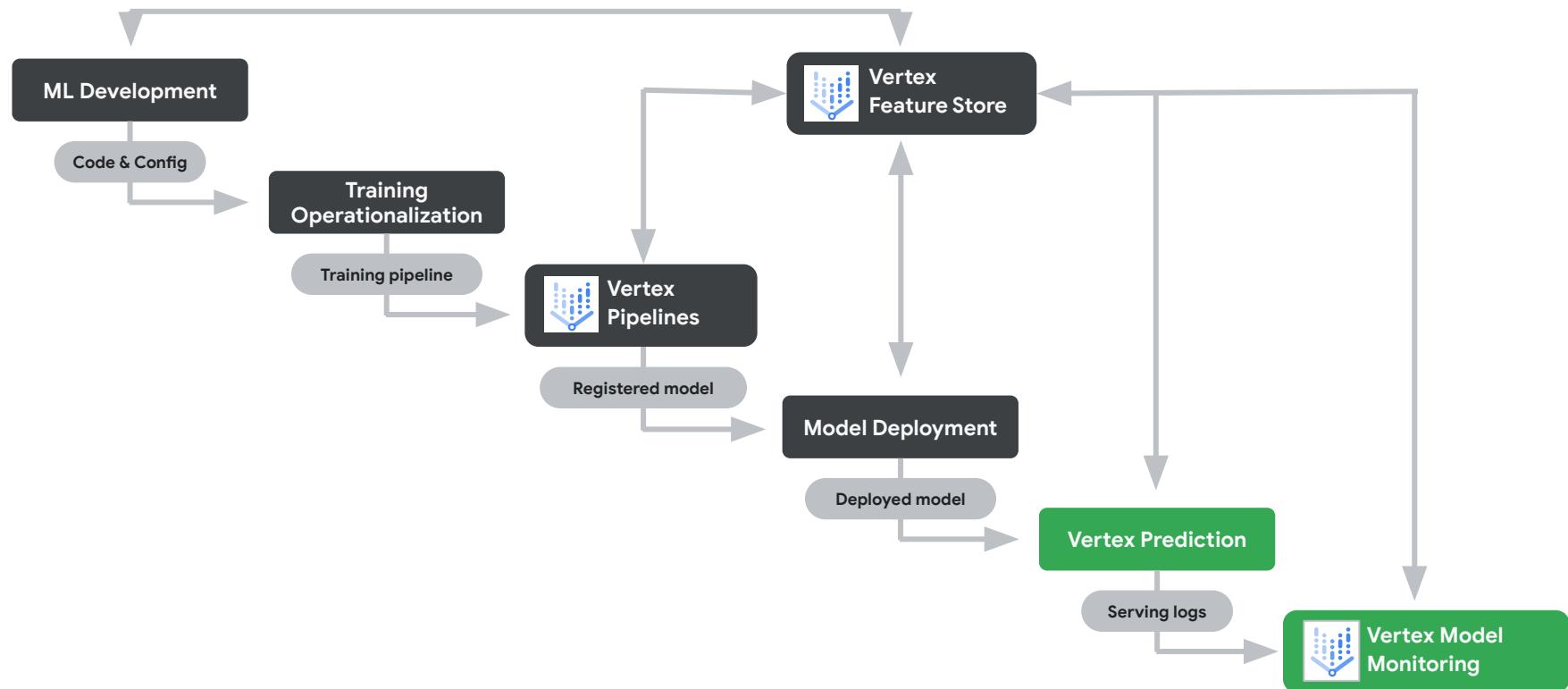


# Vertex AI

Generally available



# Vertex AIによるMLOps: Model Monitoring



# Vertex AI Model Monitoring

Preview



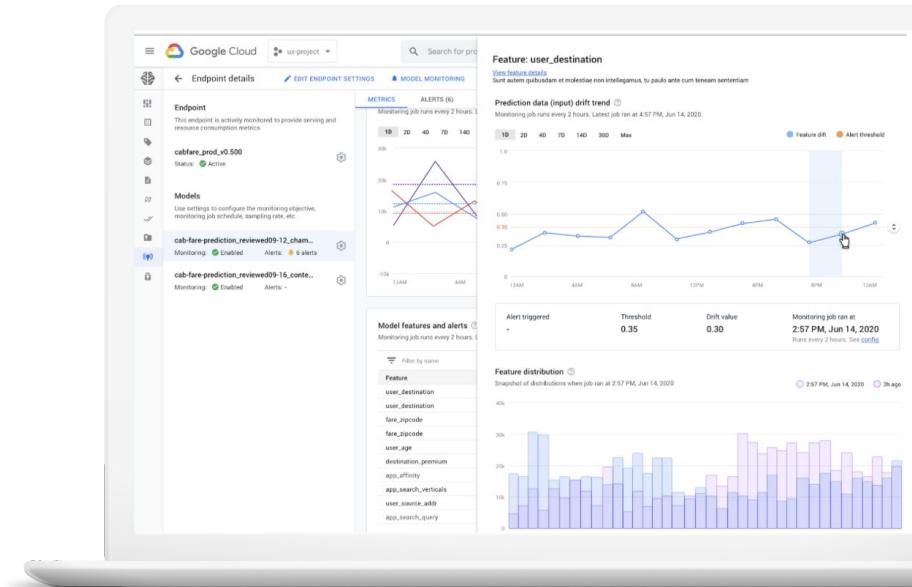
## モデル性能の監視とアラート



## モデル性能の分析

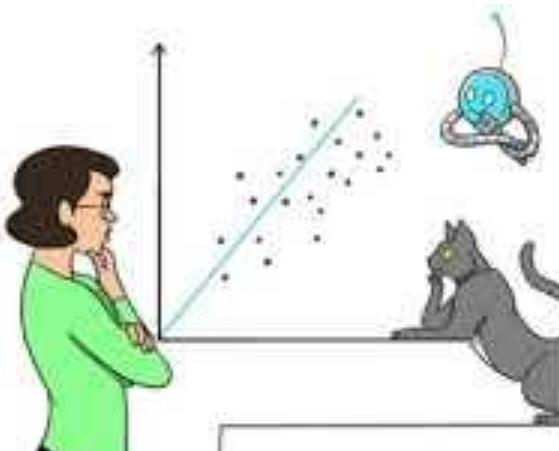


## ML パイプラインによる再学習との統合



Google Cloud

Such Training-serving Skew  
degrades the model performance.



Google

# スキュードリフトを検出

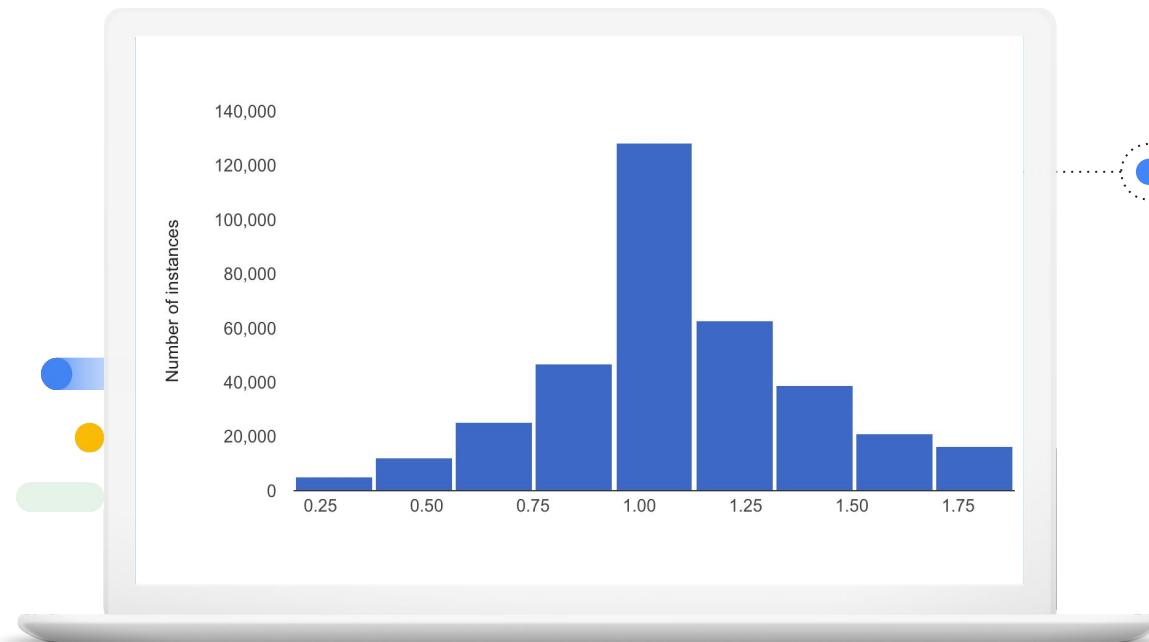
スキュー: ベースライン分布と比較

ドリフト: 経時分布と比較

分布変化の検出:

JS divergence  
(for numerical features)

L-inf distance  
(for categorical features)



# Model Monitoring による特徴量分布の監視設定

```
01 gcloud beta ai model-monitoring-jobs create
02   --project=example
03   --region=us-central1
04   --display-name=my_monitoring_job
05   --emails=example1@foobar.com,example2@foobar.com
06   --endpoint=<endpoint_id>
07   --prediction-sampling-rate=0.2
08   --feature-thresholds=feat1=0.1,feat2=0.2,feat3=0.05,feat
09   --dataset=<dataset_id>
10   --target-field=price
```

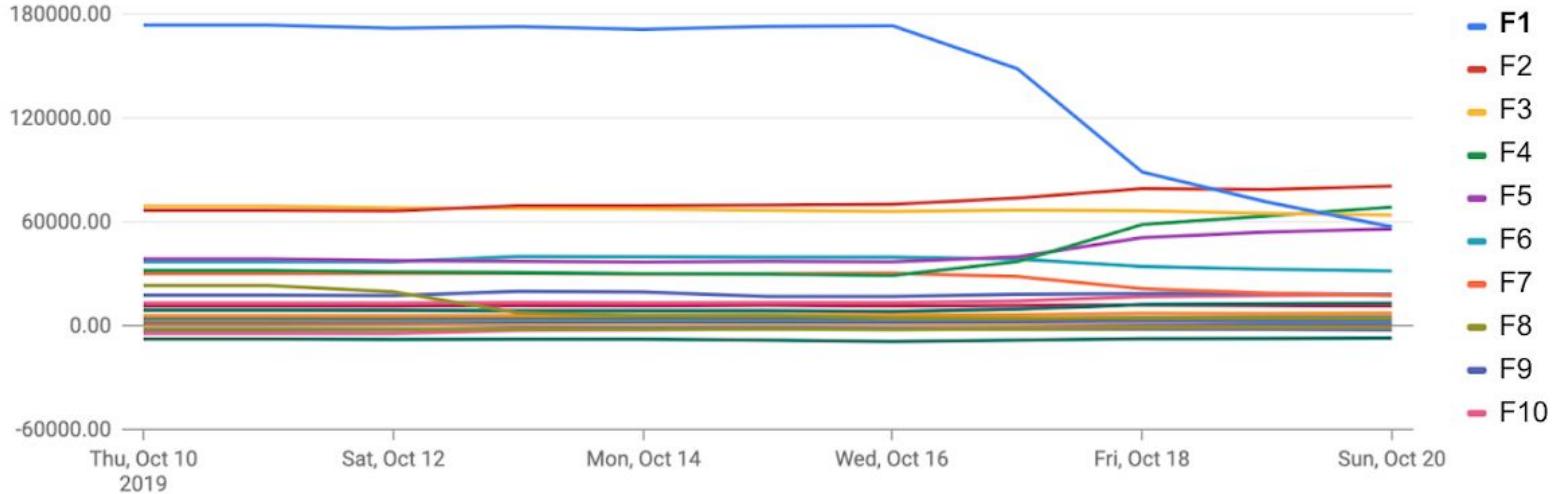
# コンソール UI での確認



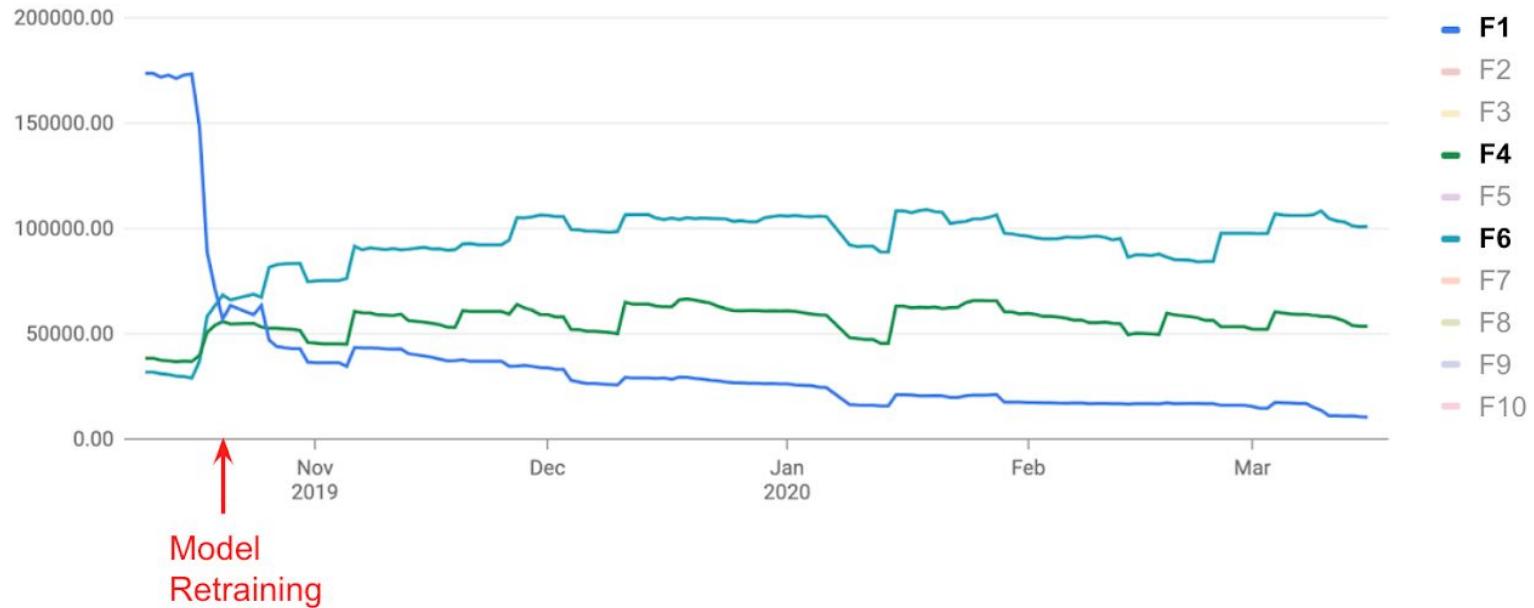
# Vertex AI Model Monitoring による特徴量重要度の監視

# Google 最大規模の ML 基盤で起きたトラブル

(Labeled - True Positives) Attribution Sum by Feature Name



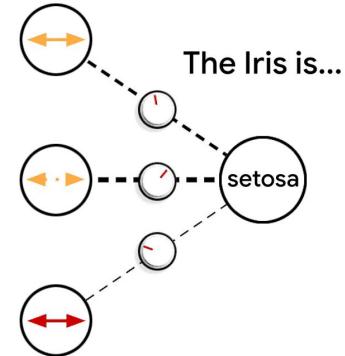
# Google 最大規模の ML 基盤で起きたトラブル



# 「正解」が得られない ML モデルには：特徴量重要度の監視

## 重要度監視のメリット

- 多次元ベクトルや embeddings の監視も可能
- 複数特徴量の監視が可能
- その他、推論サービスの不具合を検出



特徴量重要度は ML モデルが  
いまどのように動いているかを示す

# Model Monitoring による特徴量重要度の監視設定

```
01 gcloud beta ai model-monitoring-jobs create
02   --project=example
03   --region=us-central1
04   --display-name=my_monitoring_job
05   --emails=example1@foobar.com,example2@foobar.com
06   --endpoint=<endpoint_id>
07   --prediction-sampling-rate=0.2
08   --feature-thresholds=feat1=0.1,feat2=0.2,feat3=0.05,feat
09   --feature-attribution-thresholds=feat1=0.1,feat2=0.2,feat
```

**New endpoint**

Define your endpoint  
 Model settings **Model monitoring**  Monitoring objectives

**CREATE** **CANCEL**

---

**Explainability options**

In Vertex AI, models are made explainable through feature attribution, which tells you how much each feature contributed to the predicted result. You can use this information to verify that the model is behaving as expected, recognize bias in your models, and get ideas for ways to improve your model and your training data. Explainability will incur a minor additional cost. [Learn more](#)

**Select a feature attribution method**

Your model's data type determines which attribution methods are available to use. [Learn more about attribution methods](#)

None  
 Sampled Shapley (for tabular models)  
 Integrated gradients (for tabular models)  
 Integrated gradients (for image classification models)  
 XRAI (for image classification models)

**Sampled Shapley**

Assigns credit for the outcome to each feature and considers different permutations of the features. Provides a sampling approximation of exact Shapley values.

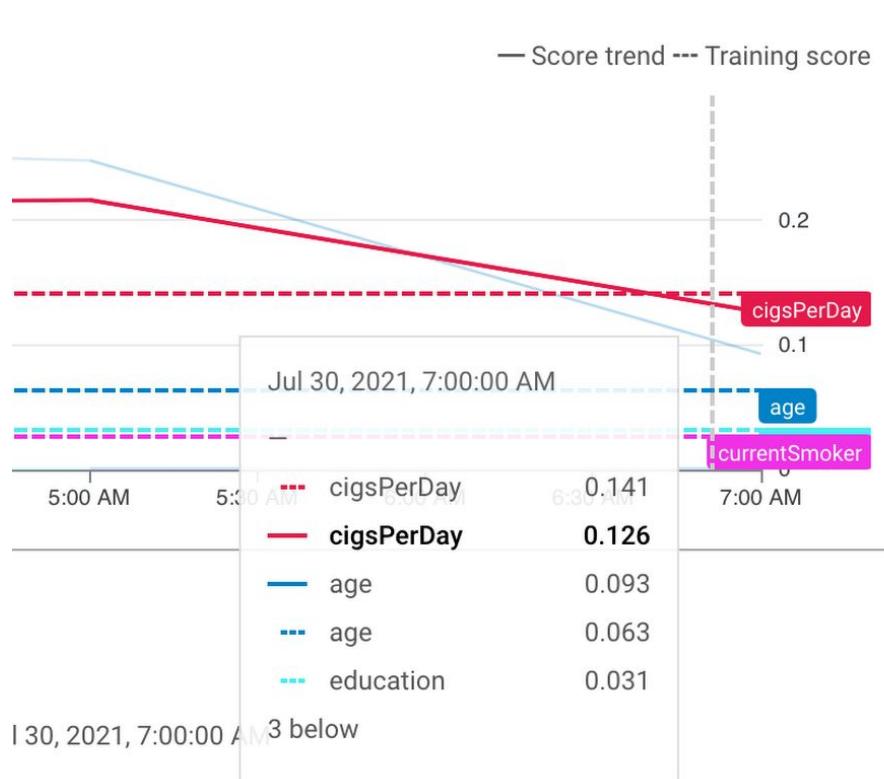
lot_size	
sq_ft	
zip_code	
built_year	

**Example**

**Path count**

The number of feature permutations to consider when approximating the Shapley values. Must be between 1 and 50.

## コンソールでの確認





# まとめ

# 機械学習サービス監視の MLOps

- 01 | ML モデルの性能は様々な要因で低下する  
→ ML モデルの現在の性能を知る「監視」が重要
- 02 | 特徴量の分布はスキューやドリフトで変化する  
→ 特徴量分布の監視で検出する
- 03 | 特徴量分布の監視だけでは不足するケースもある  
→ 特徴量重要度の監視で検出する

# Vertex AI Model Monitoring 参考資料

- Vertex AI Model Monitoring ドキュメント

<https://cloud.google.com/vertex-ai/docs/model-monitoring>

- ブログ記事「Vertex Model Monitoring で活用する、Google の MLOps 監視手法」

<https://cloud.google.com/blog/ja/topics/developers-practitioners/monitor-models-training-serving-skew-vertex-ai>

- ブログ記事「Feature Attributions の監視: Google はいかに大規模な ML サービスの障害を乗り越えたのか」

<https://cloud.google.com/blog/ja/topics/developers-practitioners/monitoring-feature-attributions-how-google-saved-one-largest-ml-services-trouble>

# Thank you.

