Machine learning model serving landscape

Machine learning model serving landscape

I am Srimugunthan, work as Data engineer in Traveler-profile team headed by Sharim Chua

TALK AGENDA:

- **Introduction to Model serving**
- **Model serving options**
- **Best practices for ML engineering**

TALK AGENDA:

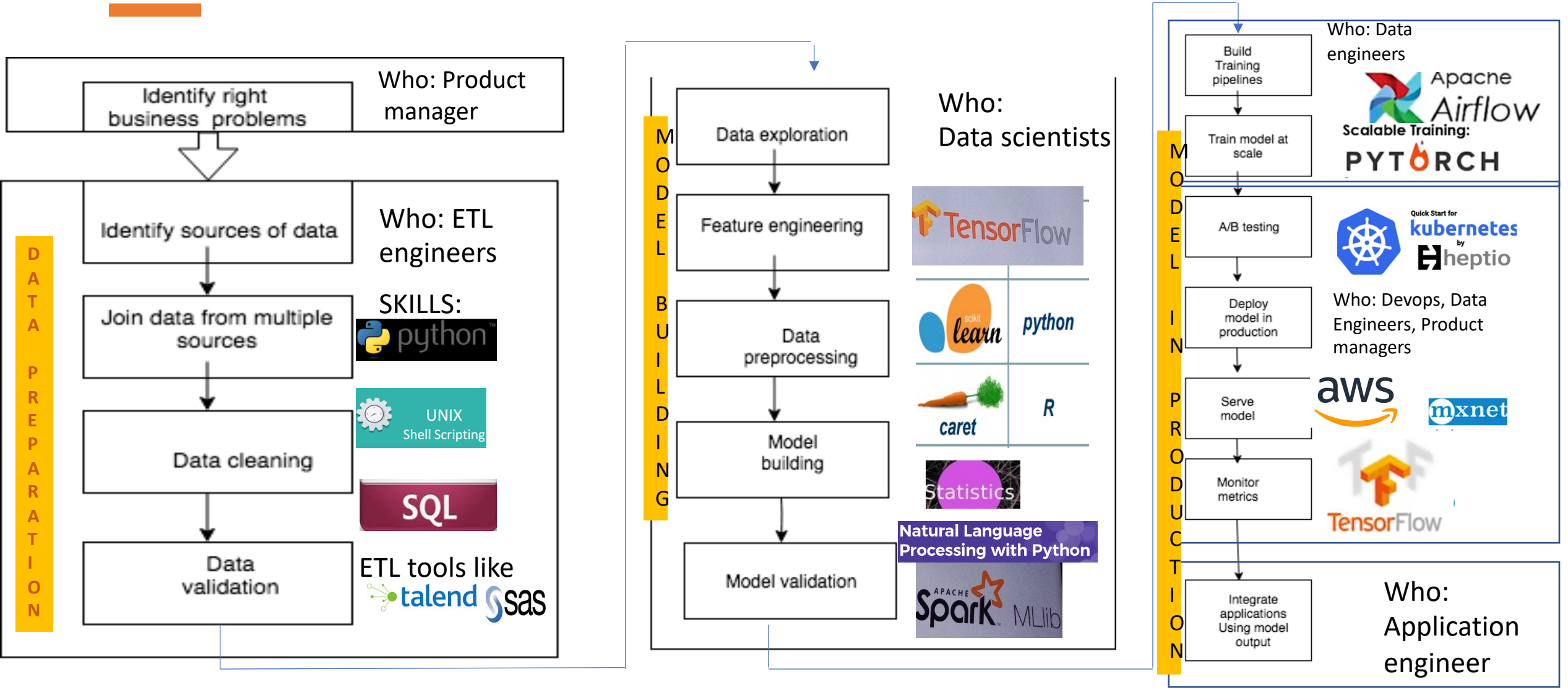
- **Introduction to Model serving**
- **Model serving options**
- **Best practices for ML engineering**

Common ML problems and models used

Problem	Models used
Supply and demand forecast	Support vector machines, ARIMA models
Price optimization	Regression trees, Logistic regression, generalized linear models
Text sentiment analysis	Naïve bayes classifier
Customer segmentation	K-means clustering, Guassian mixture models
Intrusion detection	Similarity matching algorithms, K-nearest neigbours
Churn prediction	Gradient boosted decision trees
Product Recommendation	Collaborative filtering, Matrix factorization
Fraud detection	Rule based classifiers, SVM
Spam(also defect) detection	Naïve bayes classifier
Assistive diagnosis	Random forest, Adaboost,Convolutional neural networks
Object detection	Convolutional neural network
Chatbots	Recurrent neural network

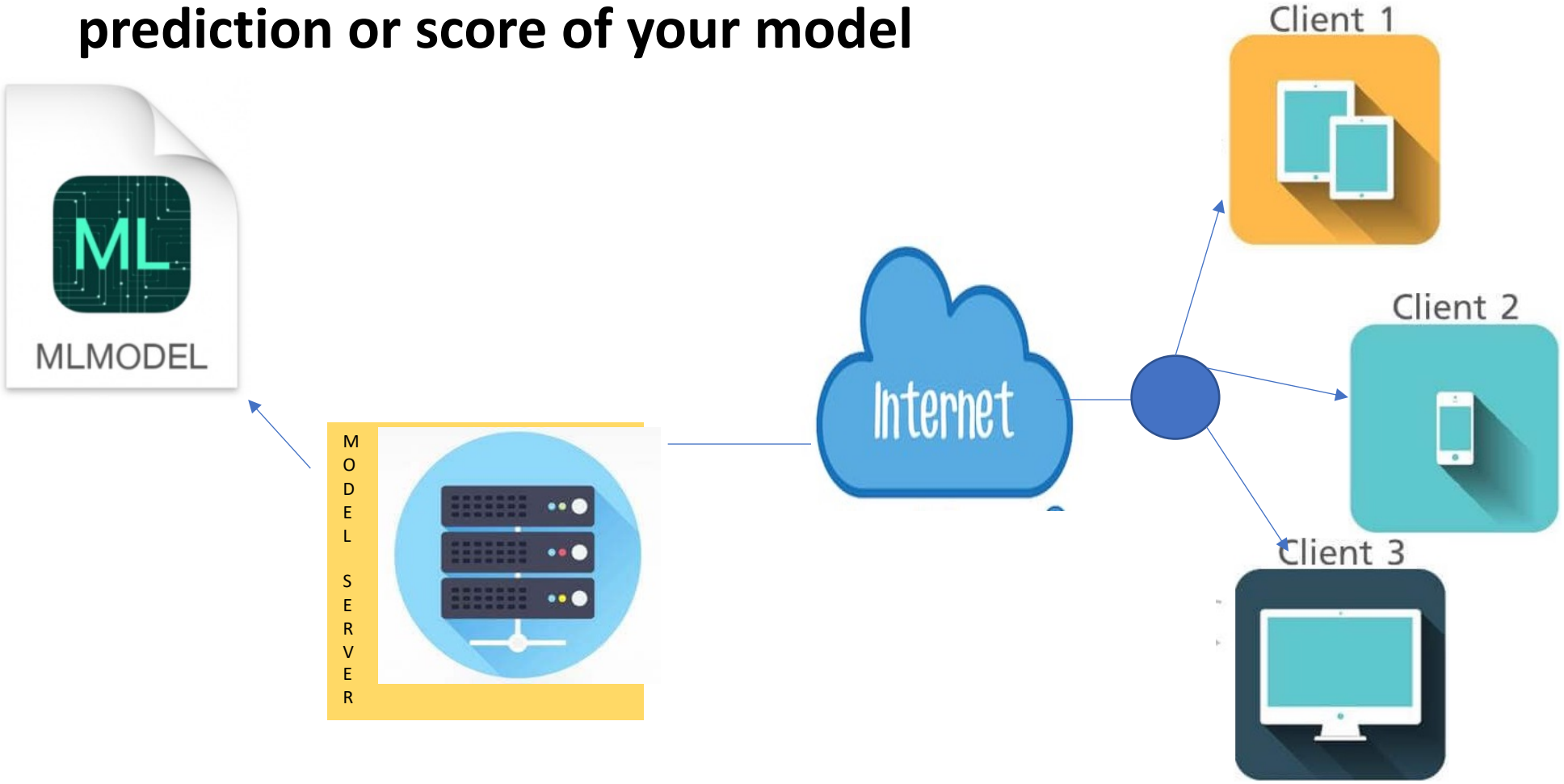
[1] “Top 10 algorithms in data mining”: http://www.cs.umd.edu/~samir/498/10Algorithms-08.pdf
[2] “The unreasonable effectiveness of data” https://research.google.com/pubs/archive/35179.pdf

Typical ML workflow



What is model serving?

- **Expose a REST endpoint which will give your recommendation, prediction or score of your model**



Distributing models

- **Bridging gap between Model developers and Deployment engineers:**
  - **Pre-materialise your model output (predictions) and store it in a DB with a REST API serving from the DB.**
  - **Package model code (Python's Pickle)**
    - **Disadvantages: Env dependent:** version mismatch between libraries when you pickle and unpickle..
  - **Package your model, code, data, dependencies into a docker container and distribute containers**
  - **Model as Data: Distribute encoded representation of trained model.**
    - **Pmml, Pfa (Portable format for analytics)**
    - **Advantages:**
      - **Code independent**
    - **Disadvantages:**
      - **Reproducibility**
      - **Some models cant be encoded(Eg: No pmml for Matrix factorization model)**
  - **Serialize your model into a common format.**

5 min demo

Use Flask to serve a machine learning model as RESTful API

But…

Model serving needed functionalities and features

- **Bridge skill gap. Zero recoding of models**
- **Make model serving performant with respect to throughput and latency**
- **Model versioning and multiple models**
  - **Deploy new models and later rollback to previous version**
  - **Compare two different models**
- **Handle model transitions**
  - **Seamlessly redeploy a new model online without downtime**
  - **Deploy models for newer data**
- **Production system needs debuggability, rollback and monitoring.**
  - **Monitoring:**
    - **Handle model decay. Check if model is performing right**

TALK AGENDA:

- Introduction to Model serving
- Model serving options
- Best practices for ML engineering

Model serving options

ModelDB

TensorFlow serving

PredictionIO

Clipper

H2O

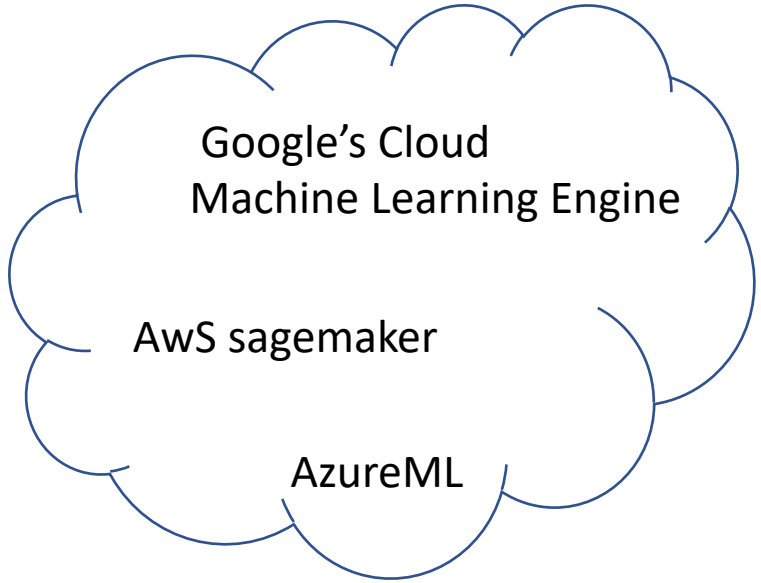
DataRobot

Skymind

Y-Hat

MXnet

MLeap



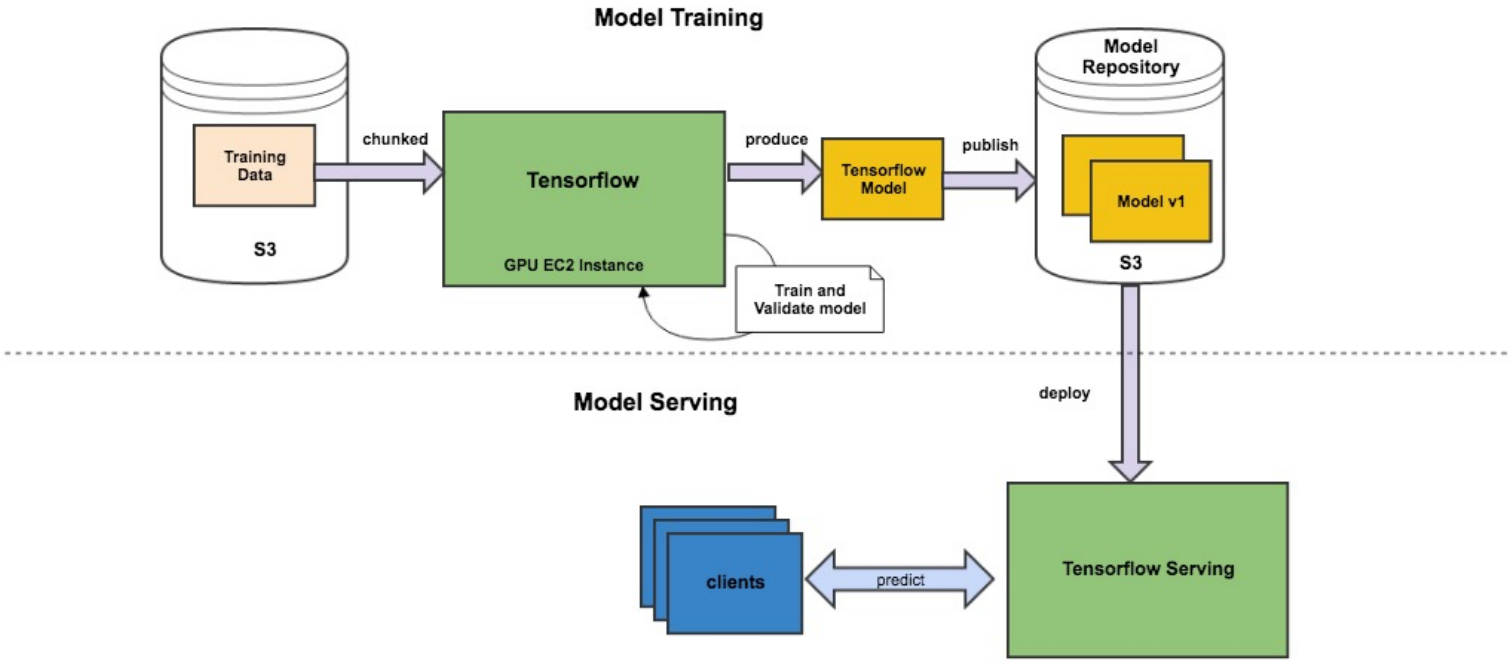
DeepDetect

Redis-ML

TensorRT

Tensor flow serving

- Tensorflow serving supports gRPC APIand RESTful API automatically.
- Supports Hot deployment of models



Zendesk model training and serving[2]

[1] https://medium.com/@yuu.ishikawa/introduction-to-restful-api-with-tensorflow-serving-9c60969b5b95
[2] https://medium.com/zendesk-engineering/how-zendesk-serves-tensorflow-models-in-production-751ee22f0f4b

MLeap

- Provides a common serialization format and execution engine for machine learning pipelines.
- serialize any PySpark/SparkML/TensorFlow pipeline into a Mleap bundle
- MLeap Bundles
  - are a graph-based, portable file format for serializing and de-serializing
  - export your bundle and load it into your Spark, Scikit-learn, or MLeap-based application
- MLeap Runtimes are running in a Java8 Virtual Machine (JVM)
- Support for model versioning is possible with Mleap+Pachyderm

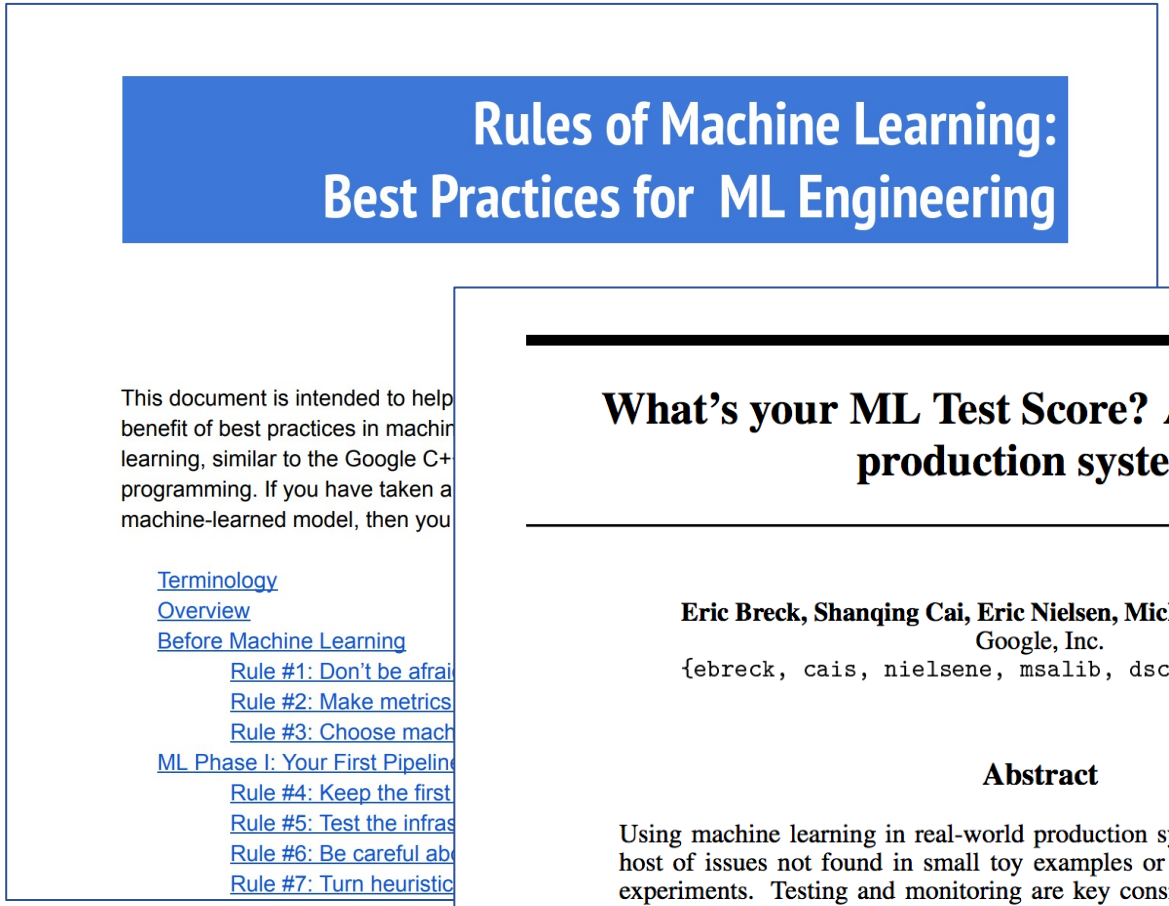
Jared besson of Hcom article: https://medium.com/hotels-com-technology/machine-learning-training-and-serving-16ab3b9c797b
https://medium.com/@combust/deploying-apache-spark-supervised-machine-learning-models-to-production-with-mleap-9e0fb57f79db

TALK AGENDA:

- **Introduction to Model serving**
- **Model serving options**
- **Best practices for ML engineering**

Google

- **Establish ML metrics first even before model is built.**
- **Continuous training and serving.**
  - **Models updated frequently (every day for some models)**
  - **Measure model staleness**
  - **Have A baseline simple model to compare**
- **Training-serving skew**
- **Privacy controls in data pipeline. GDPR deletion should propagate upto learned models.**
- **New models tested via canary process**



Rules of Machine Learning:
Best Practices for ML Engineering

This document is intended to help
benefit of best practices in machin
learning, similar to the Google C+
programming. If you have taken a
machine-learned model, then you

What's your ML Test Score? A rubric for ML production systems

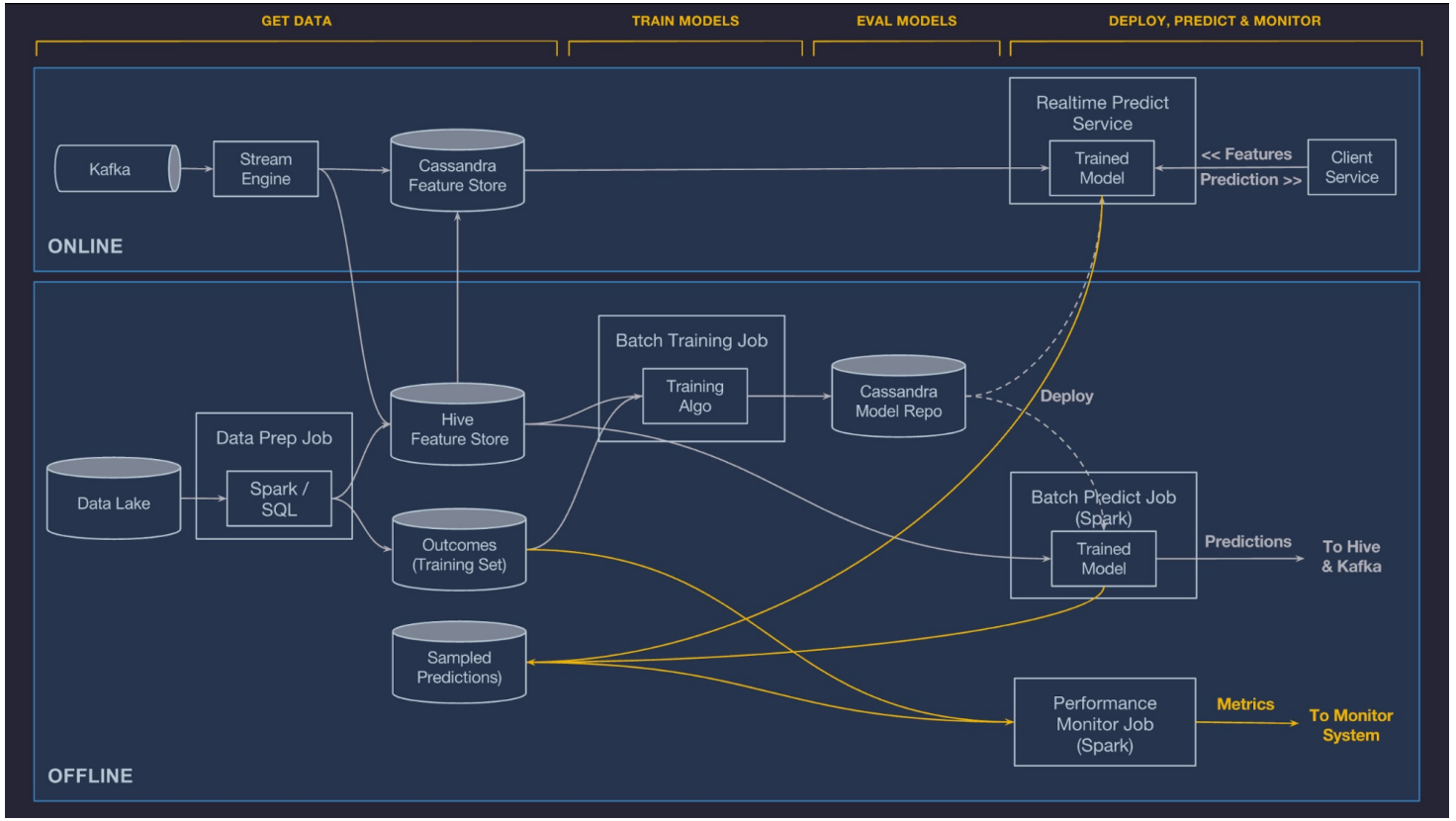
Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, D. Sculley
Google, Inc.
{ebreck, cais, nielsene, msalib, dsculley}@google.com

Abstract

Using machine learning in real-world production systems is complicated by a host of issues not found in small toy examples or even large offline research experiments. Testing and monitoring are key considerations for assessing the production-readiness of an ML system. But how much testing and monitoring is enough? We present an ML Test Score rubric based on a set of actionable tests to help quantify these issues.

Peer comparison

- **Uber**
  - Uses pen sourced components like HDFS, Spark, Samza, Cassandra, MLLib, XGBoost, and TensorFlow.
  - Centralised feature store, online and offline
  - “Before arriving at the ideal model for a given use case, it is not uncommon to train hundreds of models that do not make the cut. ”
  - **Model repository in Cassandra, with version , ownership details**
  - More than one model can be deployed at the same time to a given serving container.
    - Models identified by UUID.
    - side-by-side A/B testing of models.



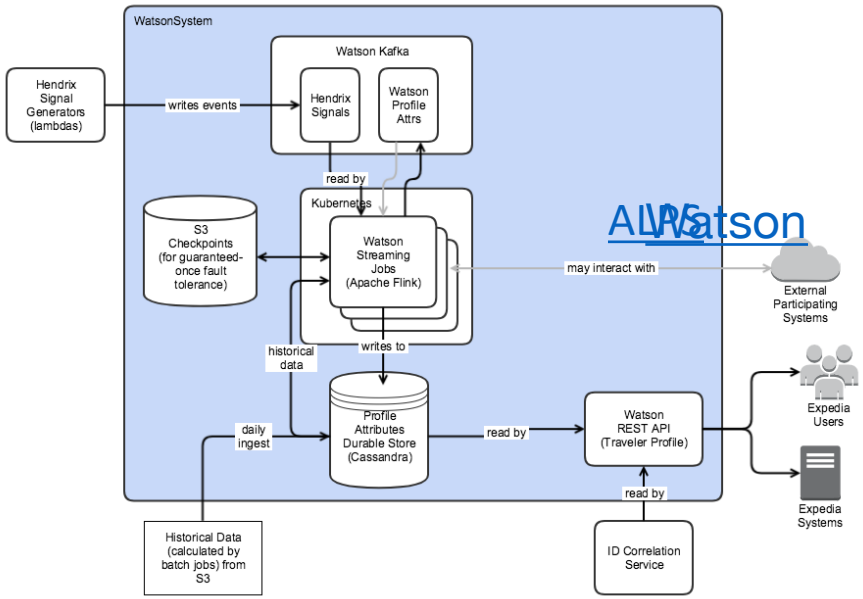
- **Booking.com**
  - **Uses H2O.ai and tensorFlow**
  - **Model serving platform**
    - **Online Near real time, Offline (automatic reconstruction of online features from offline)**
    - **Serves single and batch predictions**

Uber’s ML platform: https://eng.uber.com/michelangelo/

State of art in Expedia

MODEL DEPLOYMENT PLATFORMS:

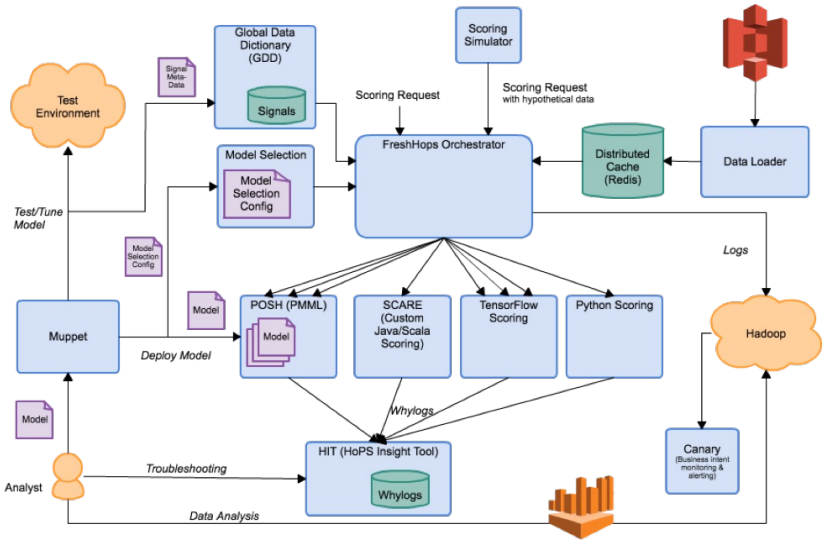
Watson



ALPS Watson

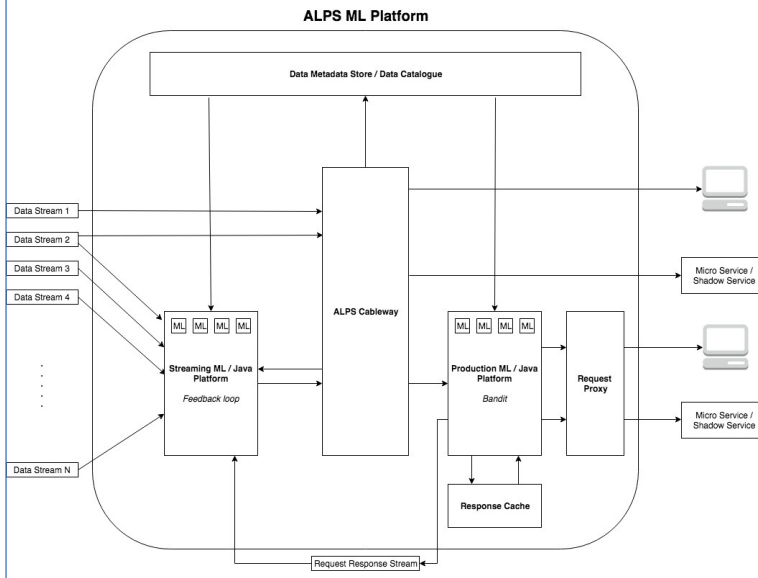
Audience Segmentation

Fresh+Hops



Ranking hotel images

ALPS



The future

- The common repository of models . Much more additions and reuse
- Model servers in the market will have more features and addons caching, monitoring, load-balancing, testing, security etc
- The input data to models comes from both real time and offline batched data
- Deploy frequently . A model stream in addition to data-stream which goes input to a ML prediction service
- A/B testing built into ML pipeline
- “Data engineers become model engineers who become responsible for ML model lifecycle”


Thanks!