



Graph AI Summit 2021

Amazon SageMaker + TigerGraph

Machine learning for every data scientist and developer

Phi Nguyen
Sr AI/ML Solution Architect

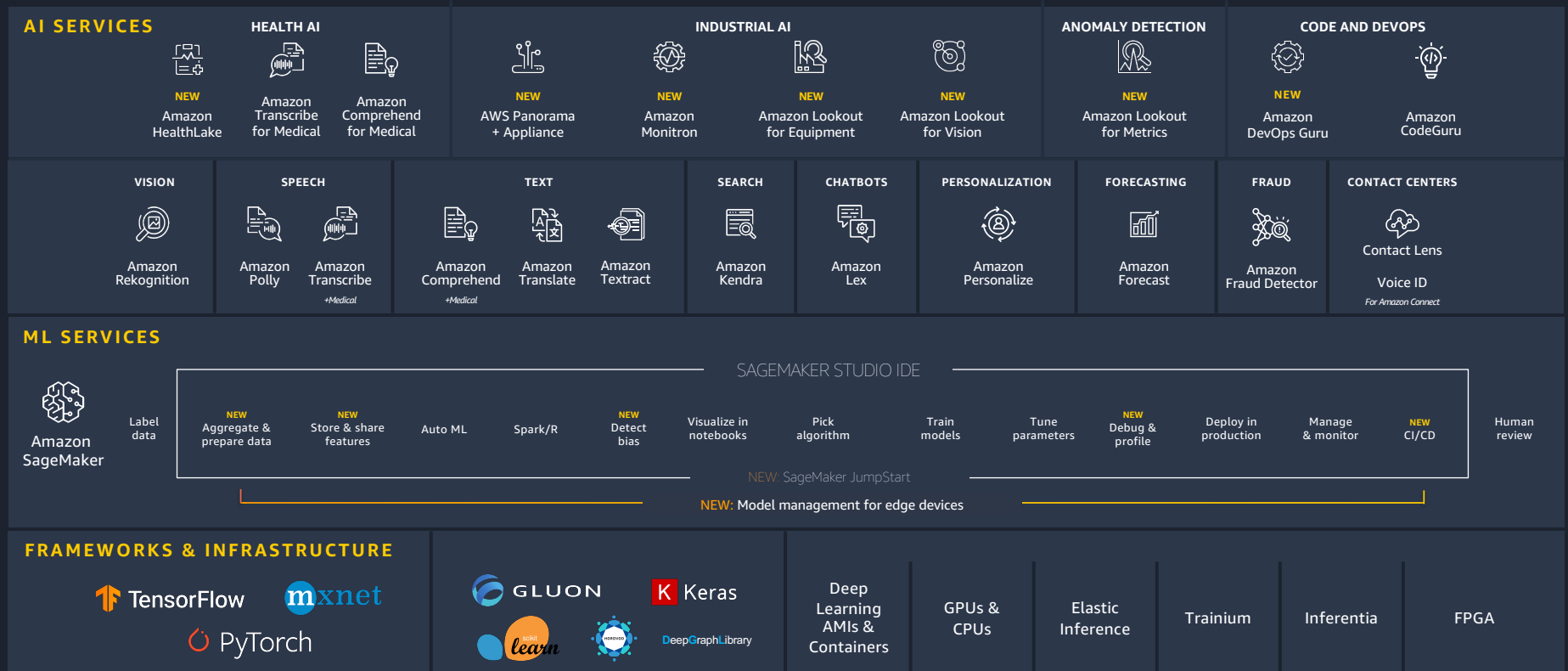
Victor Lee
VP of ML/AI

Agenda

- SageMaker Studio – Complete ML platform
- Studio Notebook
- ML on Graph
- Framework and Libraries
- Architecture Patterns

The AWS ML Stack

Broadest and most complete set of machine learning capabilities



State of machine learning



INCREASED SPENDING

By 2023, spending on AI systems will reach \$97.9B, up 2.5x from \$37.5B in 2019

—IDC



FROM PILOTING TO OPERATIONALIZING

By the end of 2024, 75% of enterprises will shift from piloting to operationalizing AI

—Gartner



ML HANDLES REAL-WORLD TASKS

Driven by advancements in GPUs and compute, availability of data, new algorithms and the cloud

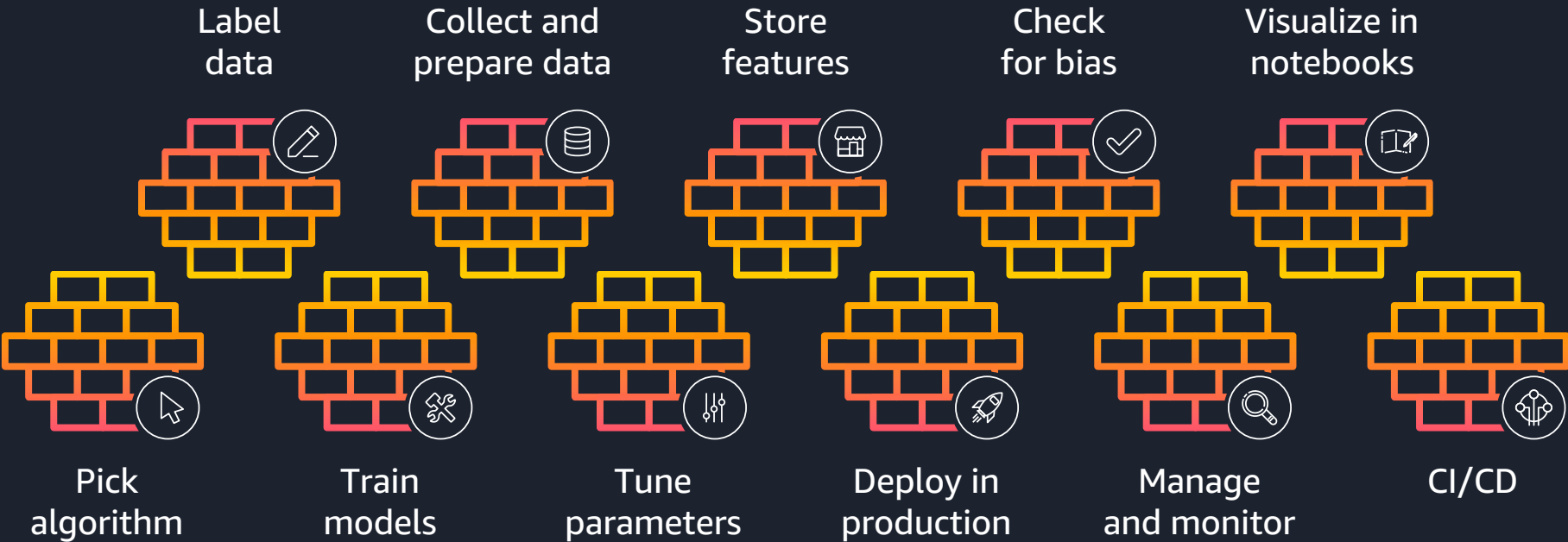


INTEGRATING ML INTO DEVOPS

ML is now part of mainstream DevOps process, not a set of specialized, isolated projects

—Gartner

Machine learning development is complex and costly



Amazon SageMaker: Built to make ML **more accessible**

SageMaker Studio IDE

Label
data



Collect and
prepare data



Store
features



Check
for bias



Visualize in
notebooks



Pick
algorithm



Train
models



Tune
parameters



Deploy in
production



Manage
and monitor



CI/CD

MODEL MANAGEMENT FOR EDGE DEVICES

Integrated Workbench

Capabilities designed specifically for ML, data preparation, experiment management, and workflows

Managed Infrastructure

Designed for ultra low latency and high throughput, automatic scaling, and distributed training

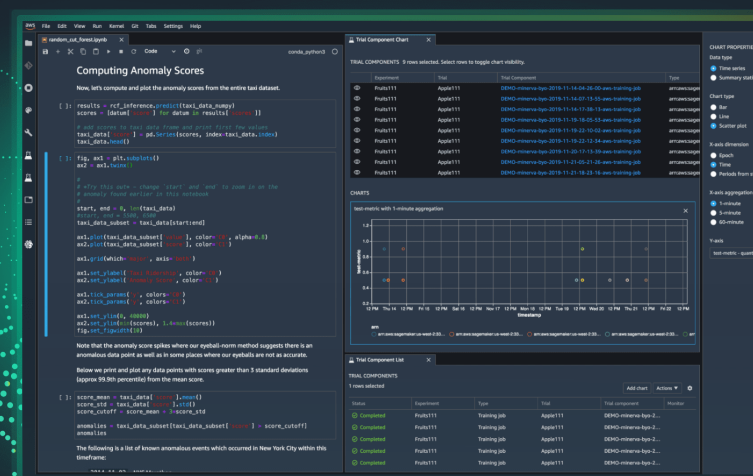
Managed Tooling

Purpose-built from the ground up to work together including auto ML, collaboration, debugger, profiler, bias analyzer, and explainability

<https://aws.amazon.com/sagemaker>

Amazon SageMaker

Most complete, end-to-end ML service



Amazon SageMaker overview

Amazon SageMaker

PREPARE

SageMaker Ground Truth

Label training data for machine learning

SageMaker Data Wrangler **NEW**

Aggregate and prepare data for machine learning

SageMaker Processing

Built-in Python, BYO R/Spark

SageMaker Feature Store **NEW**

Store, update, retrieve, and share features

SageMaker Clarify **NEW**

Detect bias and understand model predictions

BUILD

SageMaker Studio Notebooks

Jupyter notebooks with elastic compute and sharing

Built-in and Bring your-own Algorithms

Dozens of optimized algorithms or bring your own

Local Mode

Test and prototype on your local machine

SageMaker Autopilot

Automatically create machine learning models with full visibility

SageMaker JumpStart **NEW**

Pre-built solutions for common use cases

TRAIN & TUNE

Managed Training

Distributed infrastructure management

SageMaker Experiments

Capture, organize, and compare every step

Automatic Model Tuning

Hyperparameter optimization

Distributed Training Libraries **NEW**

Training for large datasets and models

SageMaker Debugger **NEW**

Debug and profile training runs

Managed Spot Training

Reduce training cost by 90%

DEPLOY & MANAGE

Managed Deployment

Fully managed, ultra low latency, high throughput

Kubernetes & Kubeflow Integration

Simplify Kubernetes-based machine learning

Multi-Model Endpoints

Reduce cost by hosting multiple models per instance

SageMaker Model Monitor

Maintain accuracy of deployed models

SageMaker Edge Manager **NEW**

Manage and monitor models on edge devices

SageMaker Pipelines **NEW**

Workflow orchestration and automation

SageMaker Studio

Integrated development environment (IDE) for ML

Amazon SageMaker key benefits

Most complete, end-to-end ML service



Accelerate ML development

20+ tools covering the entire ML development lifecycle



Boost data scientist productivity

The world's first integrated development environment (IDE)



Reduce cost

Eliminate costs of writing custom integration code with integrated functionality optimized for ML

Balancing ML agility with IT governance

ML Builders



Focus on unique business value
Self-service access
Experiment fast
Respond quickly to change



Cloud IT and DevOps



Security
Compliance
Operations
Spend management

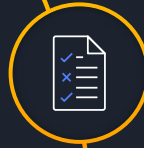
Amazon SageMaker

is devops ready



Security

Security features to help you meet strict security requirements of ML workloads



Compliance

PCI, HIPAA, SOC 1/2/3, FedRAMP, and ISO 9001/27001/27017/27018



ML workflows

Create automated workflows in minutes to support thousands of models



Scalability

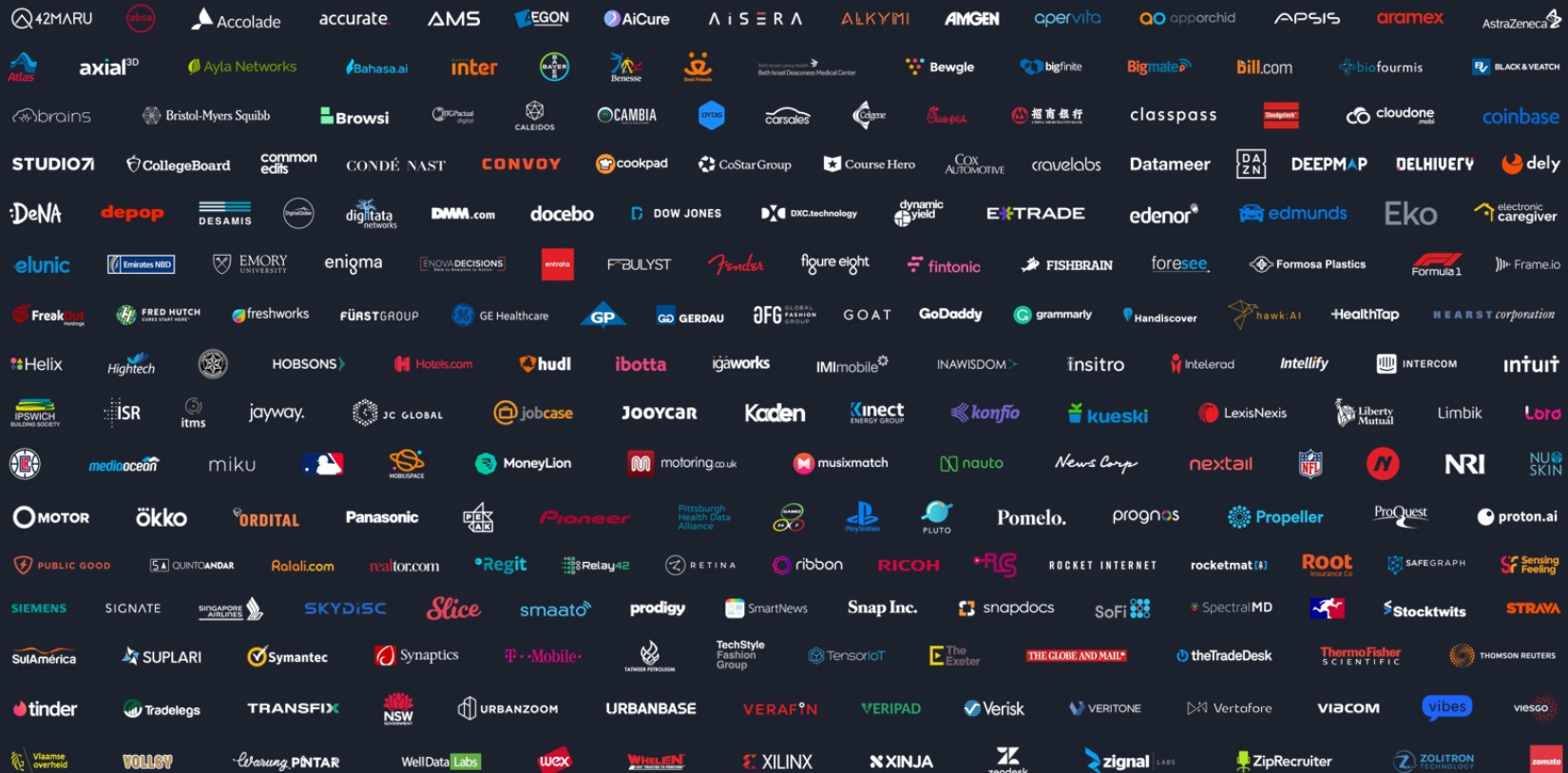
Train complex models with massive datasets



Orchestration

Automatic scheduling and execution of jobs with managed infrastructure

Tens of thousands of customers use **Amazon SageMaker**



Amazon SageMaker Studio

Amazon SageMaker Studio

Fully Integrated Development Environment (IDE) for machine learning



Collaboration at scale

Share notebooks without tracking code dependencies



Easy experiment management

Organize, track, and compare thousands of experiments



Automatic model generation

Get accurate models with full visibility and control without writing code



Higher quality ML models

Automatically debug errors, monitor models, and maintain high quality



Increased productivity

Code, build, train, deploy, and monitor in a unified visual interface

Amazon SageMaker Notebooks

Fast-start sharable notebooks



Easy access with Single Sign-On (SSO)

Access your notebooks in seconds



Fully managed and secure

Administrators manage access and permissions



Fast setup

Start your notebooks without spinning up compute resources



Easy collaboration

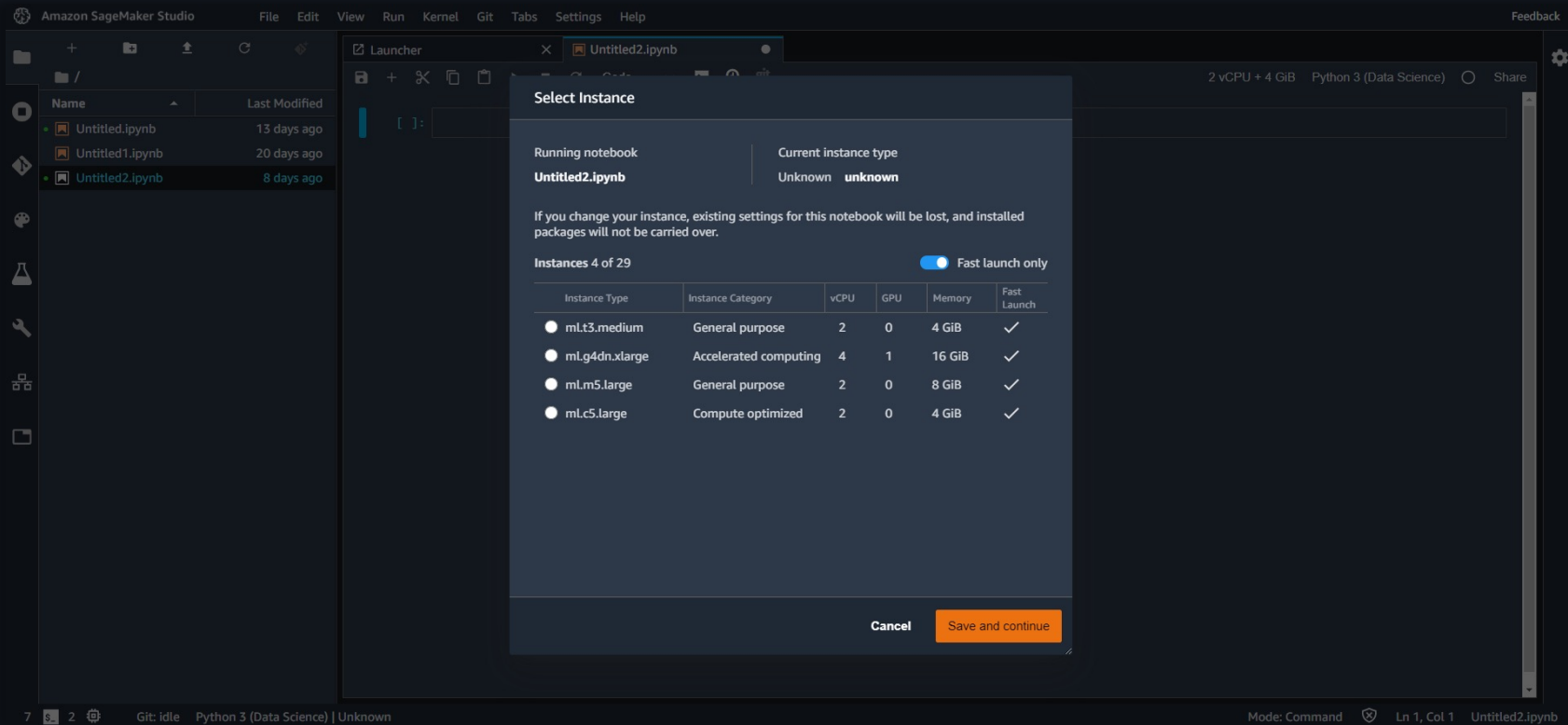
Share notebooks with a single click



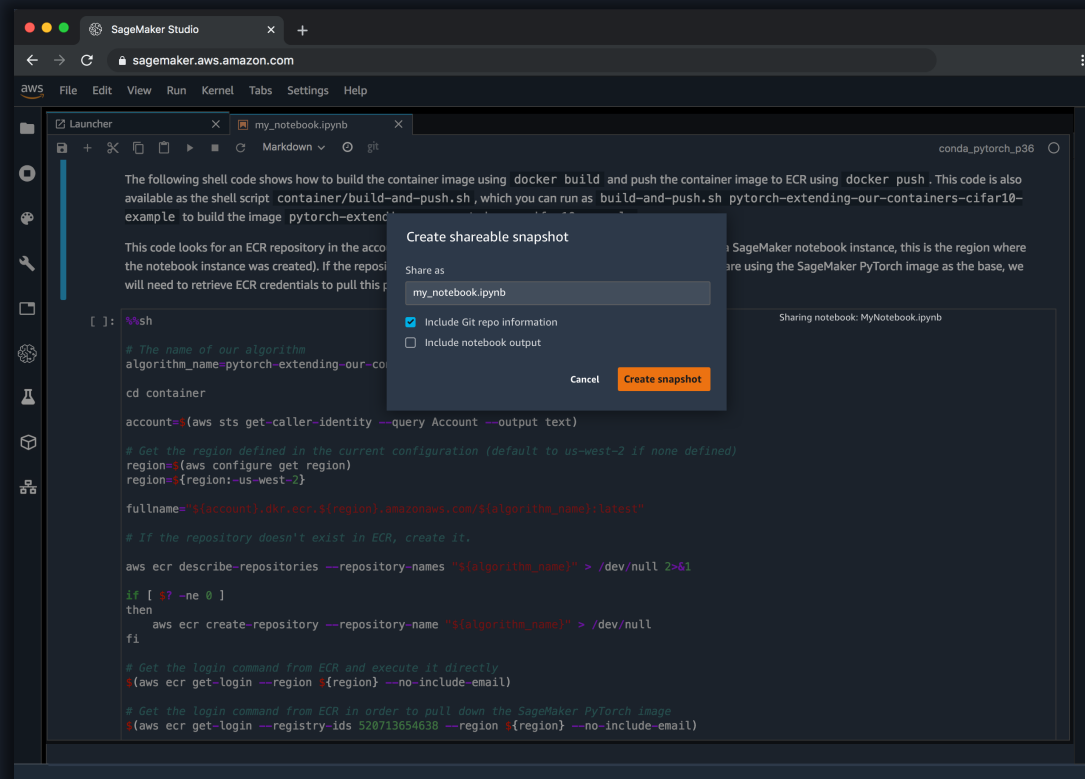
Flexible

Dial up or down compute resources (coming soon)

Use Amazon SageMaker Notebooks to easily share your work with colleagues



Code dependencies are automatically captured to enable collaboration with colleagues



Use Amazon SageMaker Studio to update models and see impact on model quality immediately

The screenshot displays the Amazon SageMaker Studio interface. The main window shows a Jupyter Notebook with the following content:

- Have the predictor variable in the first column
- Not have a header row

But first, let's convert our categorical features into numeric features.

```
[ ]: model_data = pd.get_dummies(churn)
model_data = pd.concat([model_data[ 'Churn?_True.'], model_data.drop(['Churn...
```

And now let's split the data into training, validation, and test sets. This will help prevent us from overfitting the model, and allow us to test the models accuracy on data it hasn't already seen.

```
[ ]: train_data, validation_data, test_data = np.split(model_data.sample(frac=1,
train_data.to_csv('train.csv', header=False, index=False)
validation_data.to_csv('validation.csv', header=False, index=False)
```

Now we'll upload these files to S3.

```
[ ]: boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix,
boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix,
```

On the right side, there are two panels:

- Trial Component Chart:** A line chart showing training loss over 6 periods. The y-axis is labeled 'trainloss_last' and ranges from 0.0 to 0.4. The x-axis is labeled 'period' and ranges from 0 to 6. Multiple lines represent different trials, showing a general downward trend in loss over time.
- Trial Component List:** A table listing trial components. It shows 10 rows selected. The table has columns for Status, Experiment, Type, Trial, and Trial ID.

Status	Experiment	Type	Trial	Trial ID
✓ Completed	customer-churn-predi...	Training job	Trial-3	Tra...
✓ Completed	customer-churn-predi...	Training job	Trial-2	Tra...
✓ Completed	customer-churn-predi...	Training job	Trial-1	Tra...
✓ Completed	customer-churn-predi...	Training job	Trial-0	Tra...

At the bottom of the interface, the status bar shows: Mode: Command | Ln 1, Col 1 | xgboost_customer_churn.ipynb

See pipeline execution details and metrics in real-time

Follow completed steps and monitor steps in progress

Understand the output from each step with the output logs

Monitor, change, and manage the parameters for each step

The screenshot displays the AWS SageMaker Pipeline console for a pipeline named 'TrainAbaloneModel'. The pipeline is shown as a series of steps: PreprocessAbaloneData, TrainAbaloneModel, EvaluateAbaloneModel, CheckMSEAbaloneEvaluation, and RegisterAbaloneModel. The 'TrainAbaloneModel' step is currently selected, and its details are shown in a panel on the right. The pipeline status is 'Completed' (indicated by a green dot), and it started on 2/22/2021 at 10:57 AM, taking 12m46s to complete.

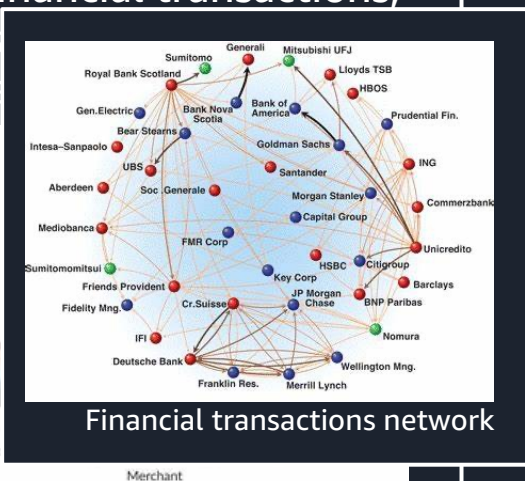
TrainAbaloneModel (Status: Completed, Started time: 2/22/2021, 10:57 AM, Elapsed time: 12m46s)

Output	Logs	Information
Metrics		Value
train:rmse		1.68371
validation:rmse		2.22083
Files		S3 bucket URI
model.tar.gz		s3://sagemaker-project-p-7k5s...

Machine Learning using Graphs

Fraud & Abuse

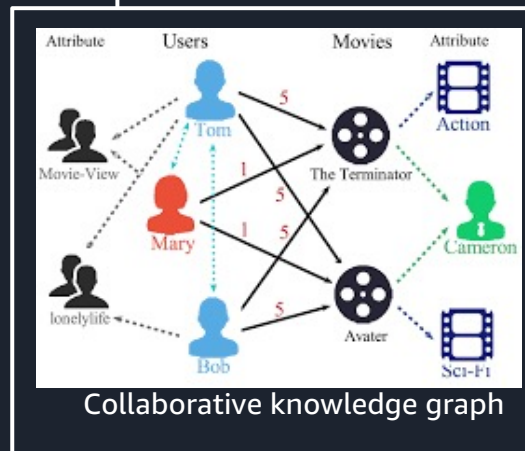
Detect malicious accounts, fraudulent financial transactions, fake reviews, insurance claims



Financial transactions network

Recommendations

Products, media, articles, experiences, jobs, courses, spouses,



Collaborative knowledge graph

to bought

How to Lie with Statistics
Darrell Huff
Illustrated by Irving Berlin

How to Lie with Statistics
Darrell Huff
★★★★☆ 478
Kindle Edition
\$6.91

Marketing

Who should get a discount? Who are the influencers? Who are the risk of churning?



Social network

Tasks in graph learning

Node classification

- Detect malicious accounts
- Target right customers

Link prediction

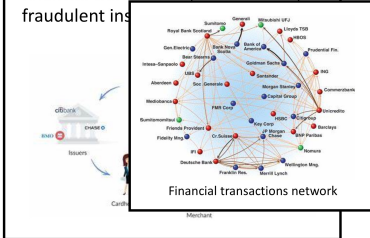
- Recommendations
- Predict missing relations in a knowledge graph

Graph classification

- Predict the property of a chemical compound

Fraud & Abuse

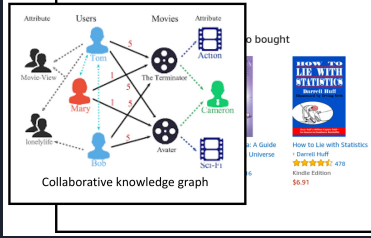
Detect malicious accounts, fraudulent financial transactions, fake reviews, fraudulent insurance claims, ...



Financial transactions network

Recommendations

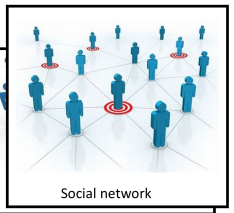
Products, media, articles, experiences, jobs, courses, spouses, ...



Collaborative knowledge graph

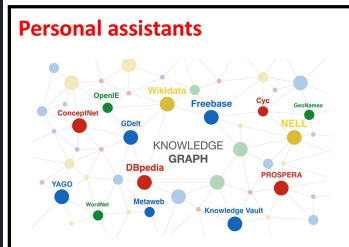
Marketing

Who should get a discount? Who are the influencers? Who are the risk of churning?



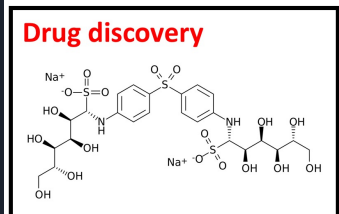
Social network

Personal assistants



KNOWLEDGE GRAPH

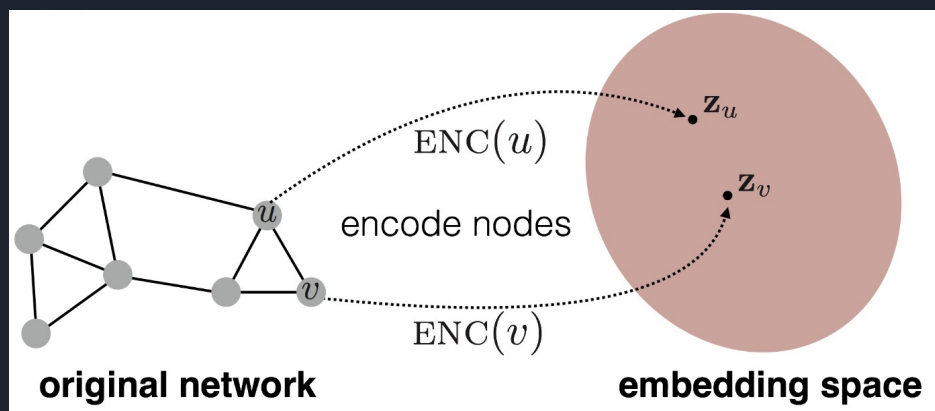
Drug discovery



Graph learning and node embeddings

Embed nodes to a low-dimension space so that these embeddings capture the essential task-specific information and use them to train off-the-self classifiers.

- For example, node similarities in the embedding space approximates the similarities in the original graph.



Representation Learning on Networks, snap.stanford.edu/proj/embeddings-www, WWW 2018

Traditional graph learning approaches

Generate embeddings by manual feature engineering

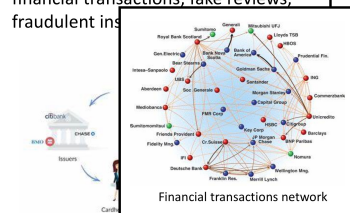
- Requires domain expertise, involves considerable manual fine-tuning, time consuming, does not scale, ...

Automatically generate embeddings using unsupervised dimensionality reduction approaches

- Singular value decomposition, tensor decomposition, co-factorization, deep walks, etc.
- Cannot effectively combine rich attributes with network structure.
- Employ mostly (multi-)liner models.
- Do not allow for end-to-end learning.

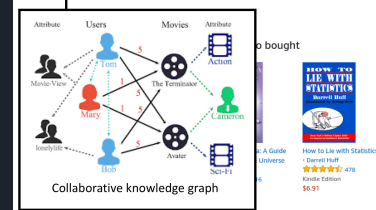
Fraud & Abuse

Detect malicious accounts, fraudulent financial transactions, fake reviews, fraudulent insurance claims, ...



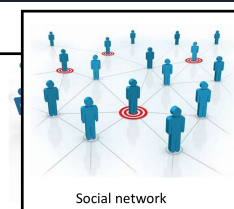
Recommendations

Products, media, articles, experiences, jobs, courses, spouses, ...

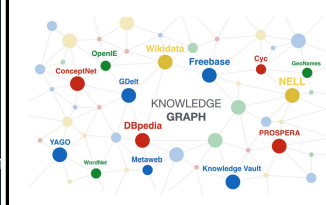


Marketing

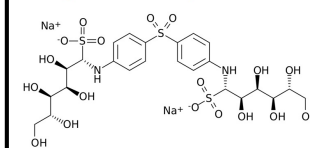
Who should get a discount? Who are the influencers? Who are the risk of churning?



Personal assistants



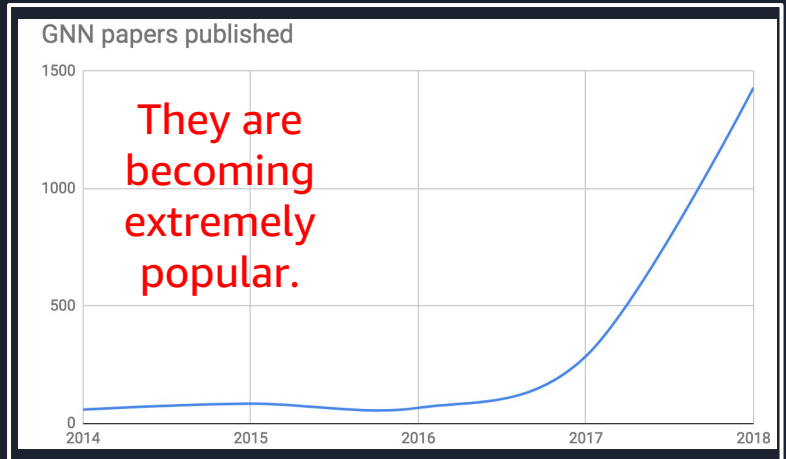
Drug discovery



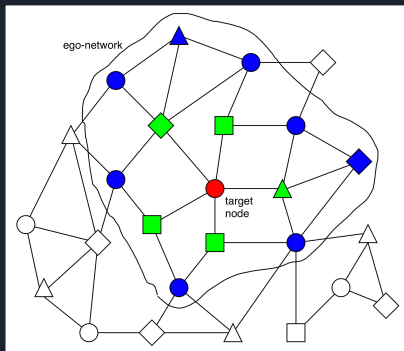
Can we do better?

Graph Neural Network (GNN)

A family of (deep) neural networks that learn node, edge, and graph embeddings.



How do GNNs work?

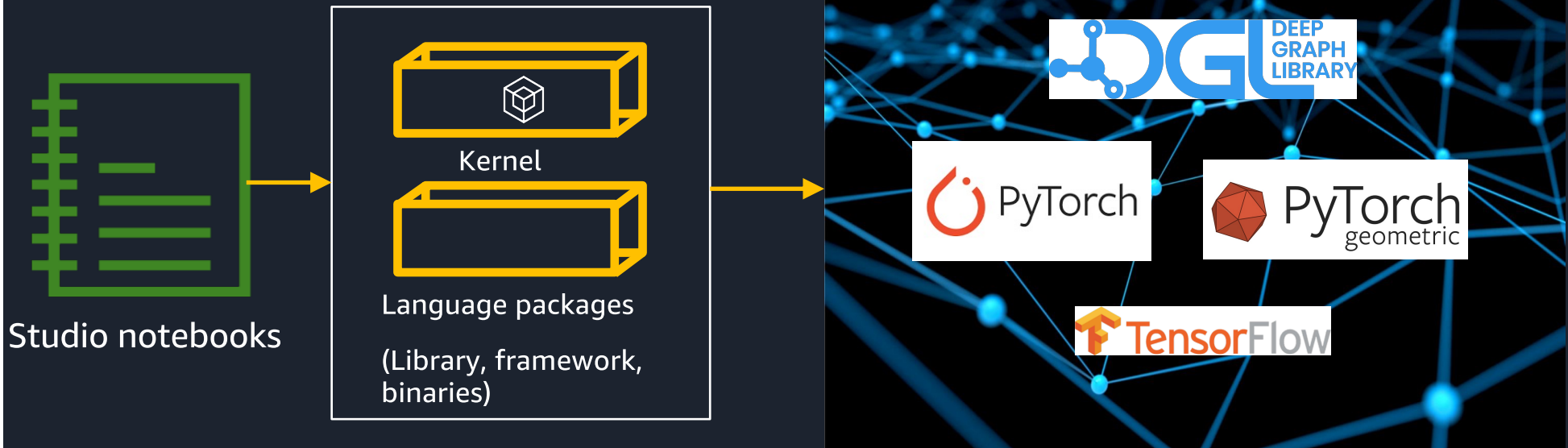


An ego-network around each node is used to learn an embedding that captures task-specific information.

The embeddings use both the structure of the graph and the features of the nodes and edges.

The embeddings are learned in an end-to-end fashion; thus, the predictions are a function of the target node's ego-network.

Framework and Libraries

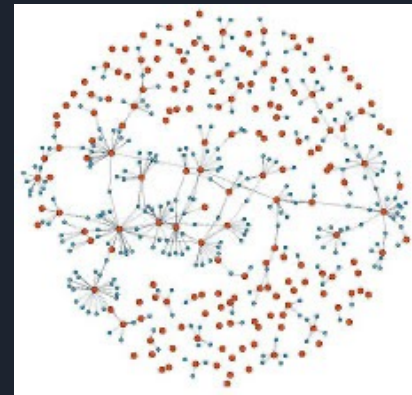


SageMaker image

SageMaker Studio + TigerGraph Cloud for feature engineering



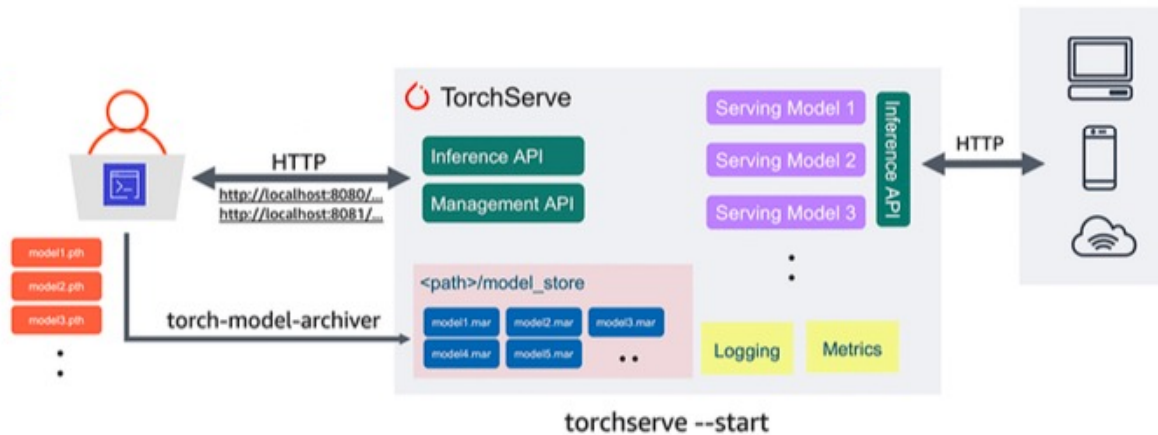
Studio notebooks



Model Deployment

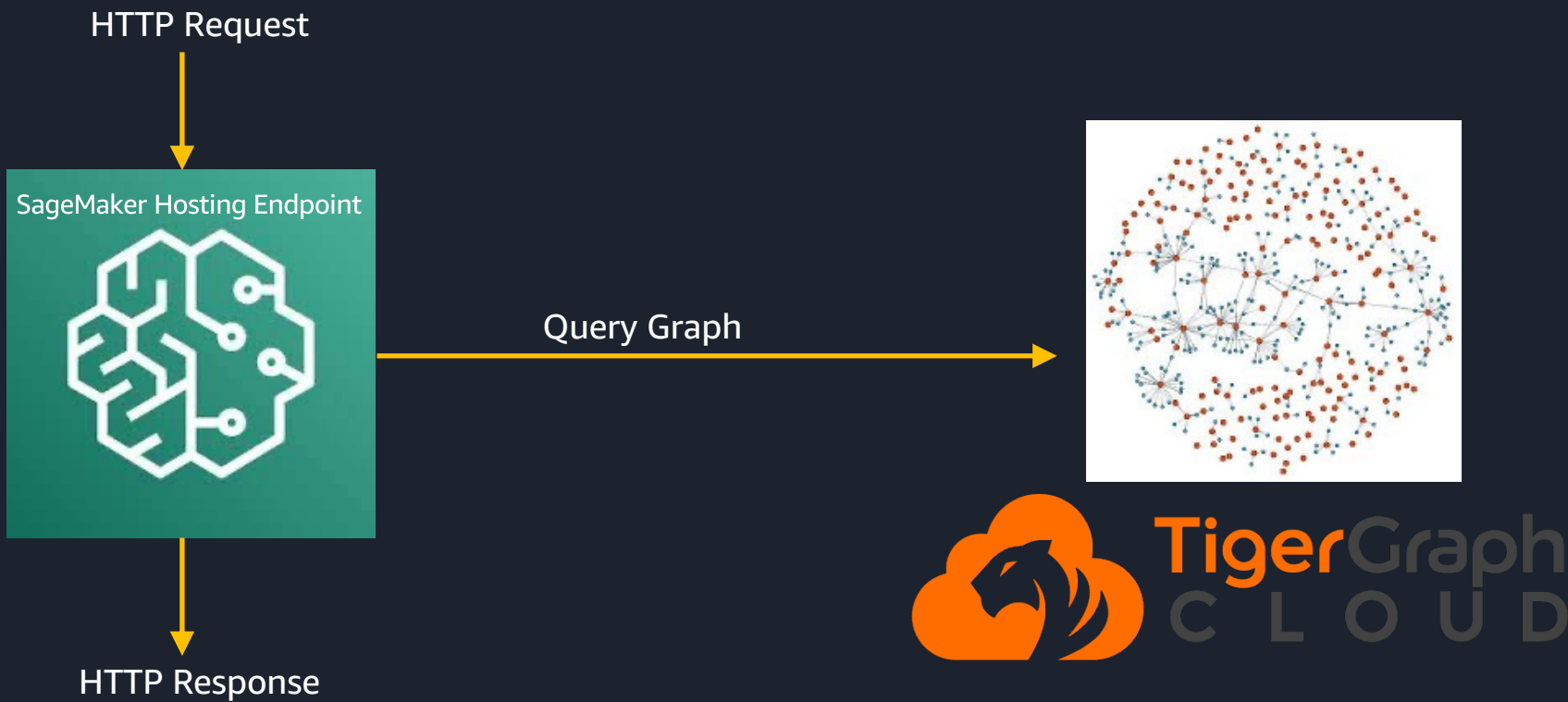


TORCHSERVE



- Default handlers for common use cases (e.g., image segmentation, text classification) along with custom handlers support for other use cases and a Model Zoo
- Multi-model serving, Model versioning and ability to roll back to an earlier version
- Robust HTTP APIS - Management, Inference and Metrics
- Ensemble support, Automatic batching, Logging and support for custom metrics
- **Sagemaker, Kubernetes, Prometheus, Kubeflow/KFServing w/canary rollouts, Captum, MLflow, Inferentia**
- (under development) C++ backend

SageMaker Studio + TigerGraph Cloud / Real time endpoint



Thank you!