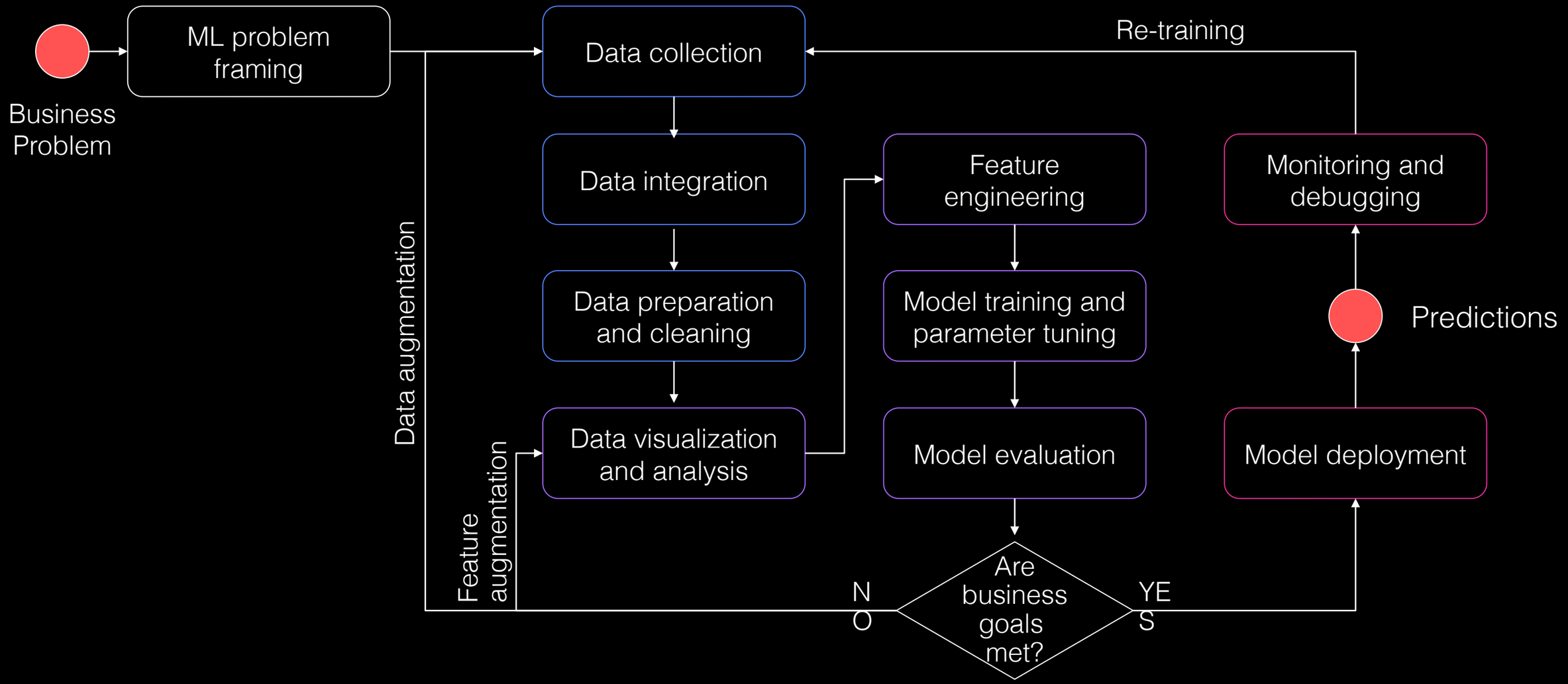
# Build, train and deploy Machine Learning models on Amazon Web Services

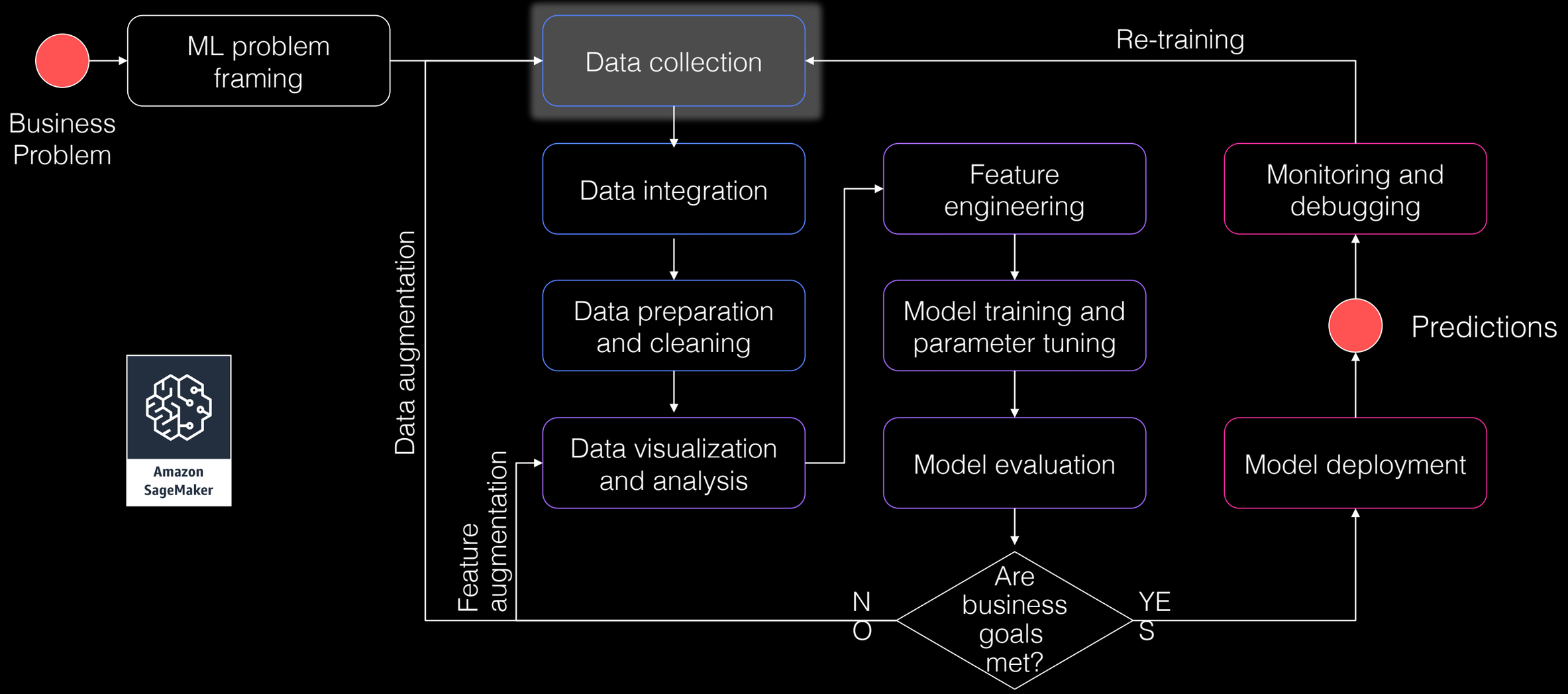
Julien Simon
Global Evangelist, AI & Machine Learning, AWS
@julsimon

# Machine learning cycle

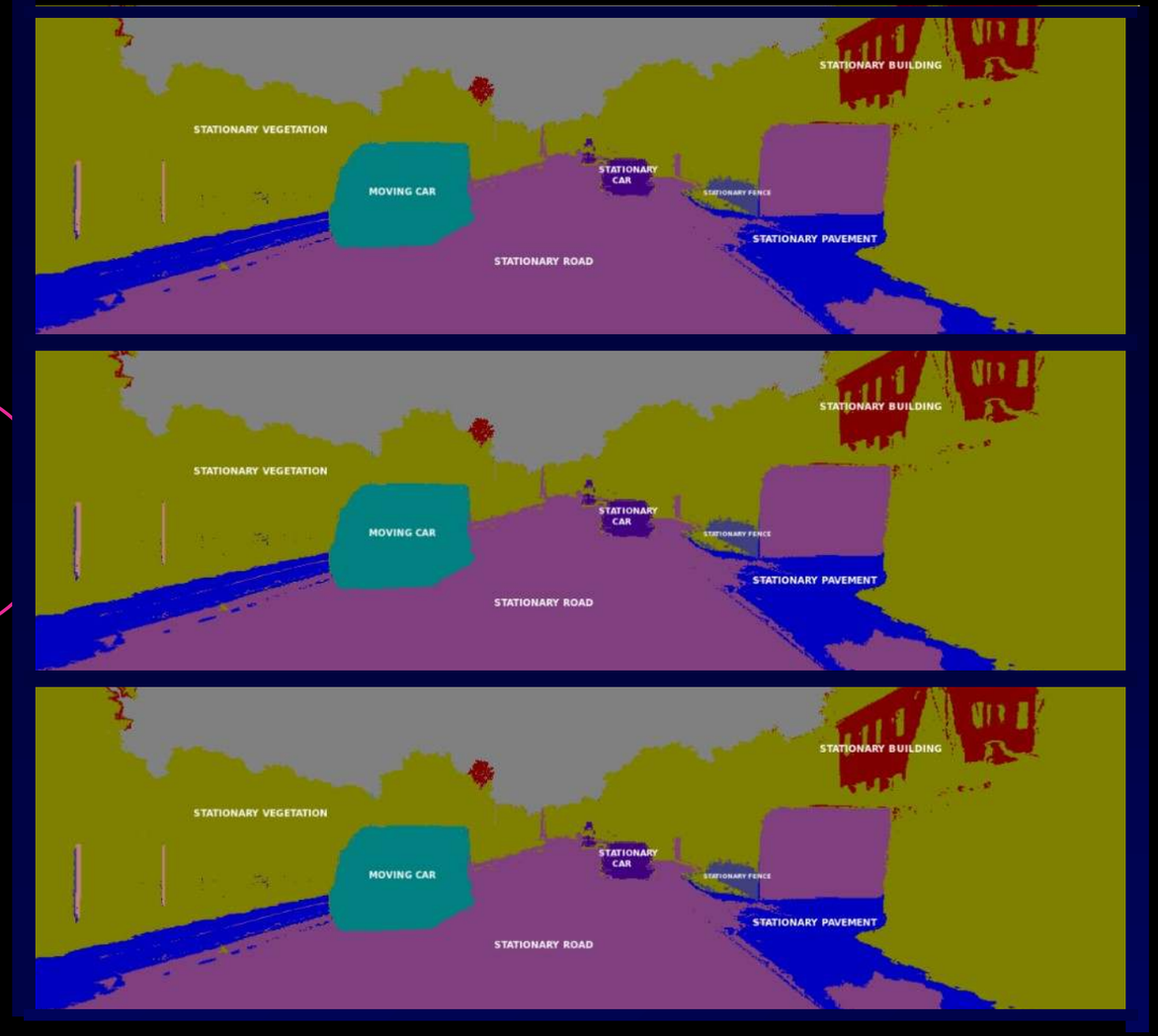


# AWS services for Machine Learning

# Build your dataset

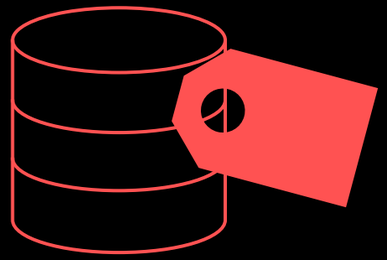


# Annotating data at scale is time-consuming and expensive

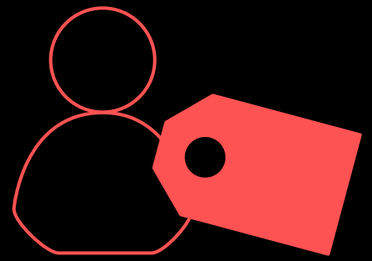


# Amazon SageMaker Ground Truth

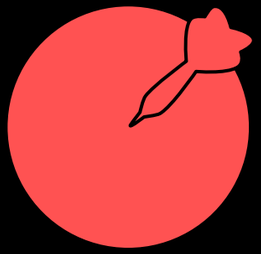
## Build scalable and cost-effective labeling workflows



Quickly label training data



Easily integrate human labelers



Get accurate results

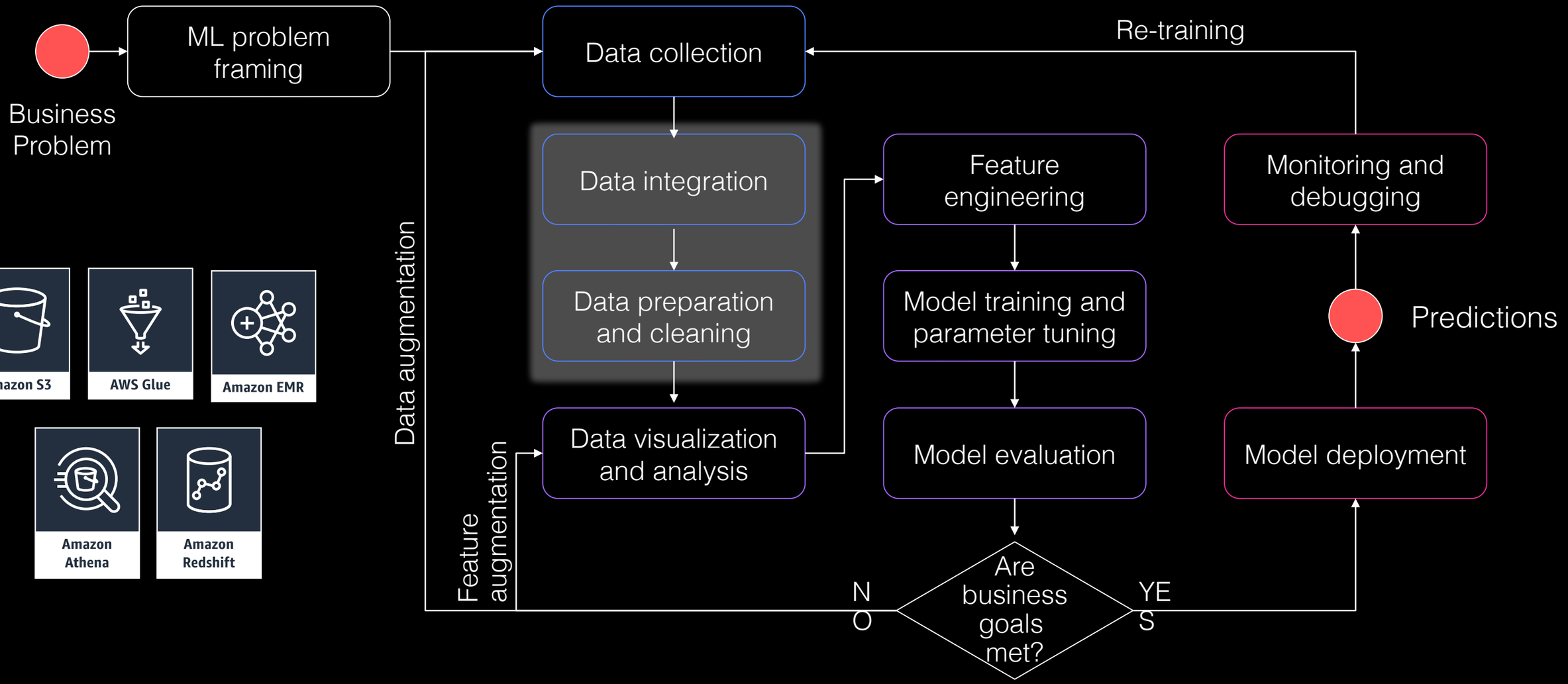
## KEY FEATURES

Automatic labeling via machine learning

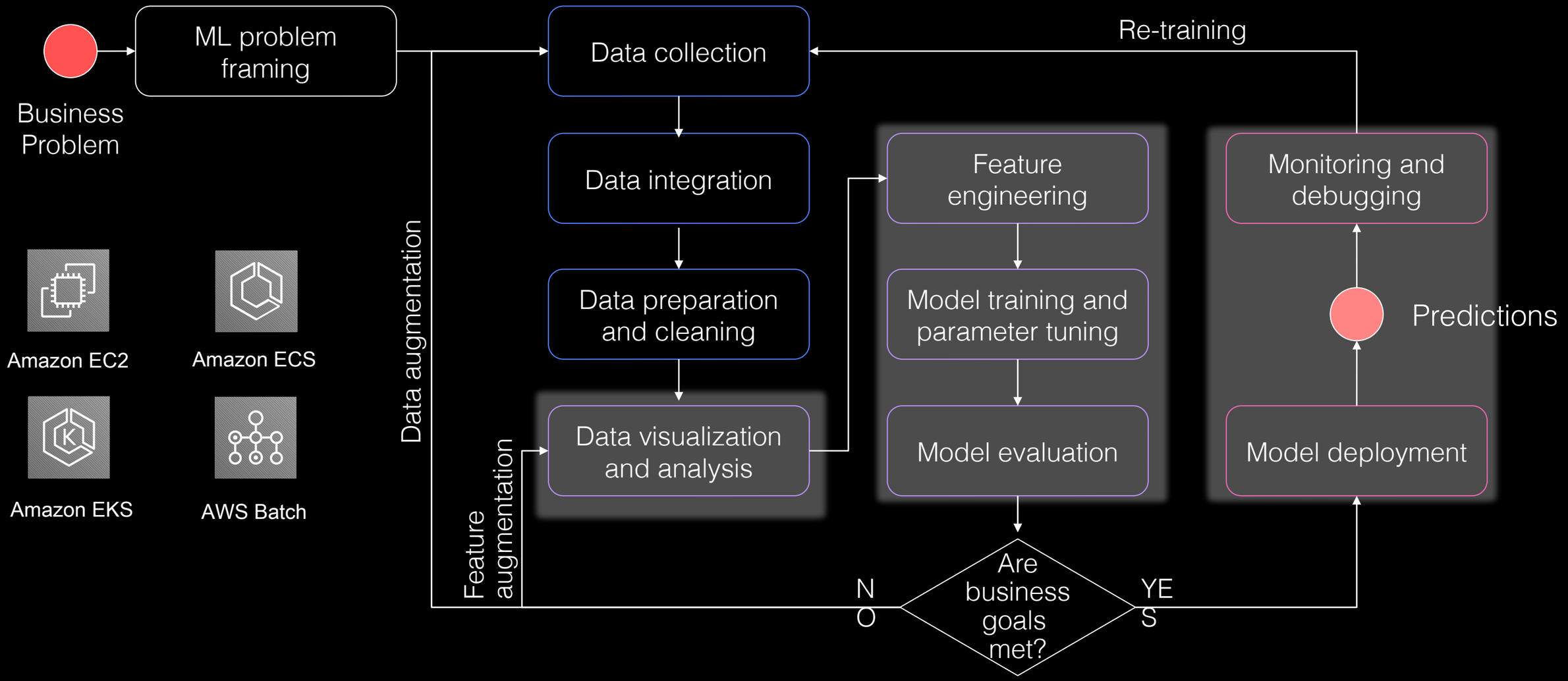
Ready-made and custom workflows for image bounding box, segmentation, and text

Private and public human workforce

# Prepare your dataset for Machine Learning



# Build, train and deploy models using compute services



# AWS Deep Learning AMIs

Preconfigured environments on Amazon Linux or Ubuntu

**Deep Learning containers for Tensorflow and Apache MXNet**

**Conda AMI**

For developers who want pre-installed pip packages of DL frameworks in separate virtual environments.

**Base AMI**

For developers who want a clean slate to set up private DL engine repositories or custom builds of DL engines.



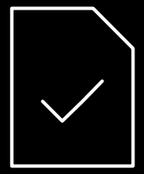




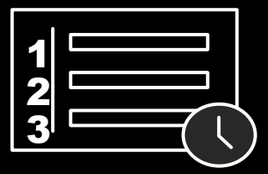



# Amazon SageMaker

# Amazon SageMaker

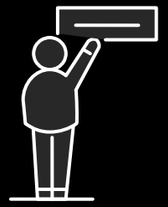


Collect and prepare training data



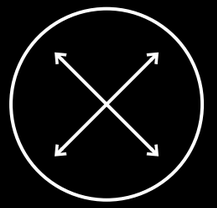

Choose and optimize your ML algorithm

Set up and manage environments for training



Train and Tune ML Models

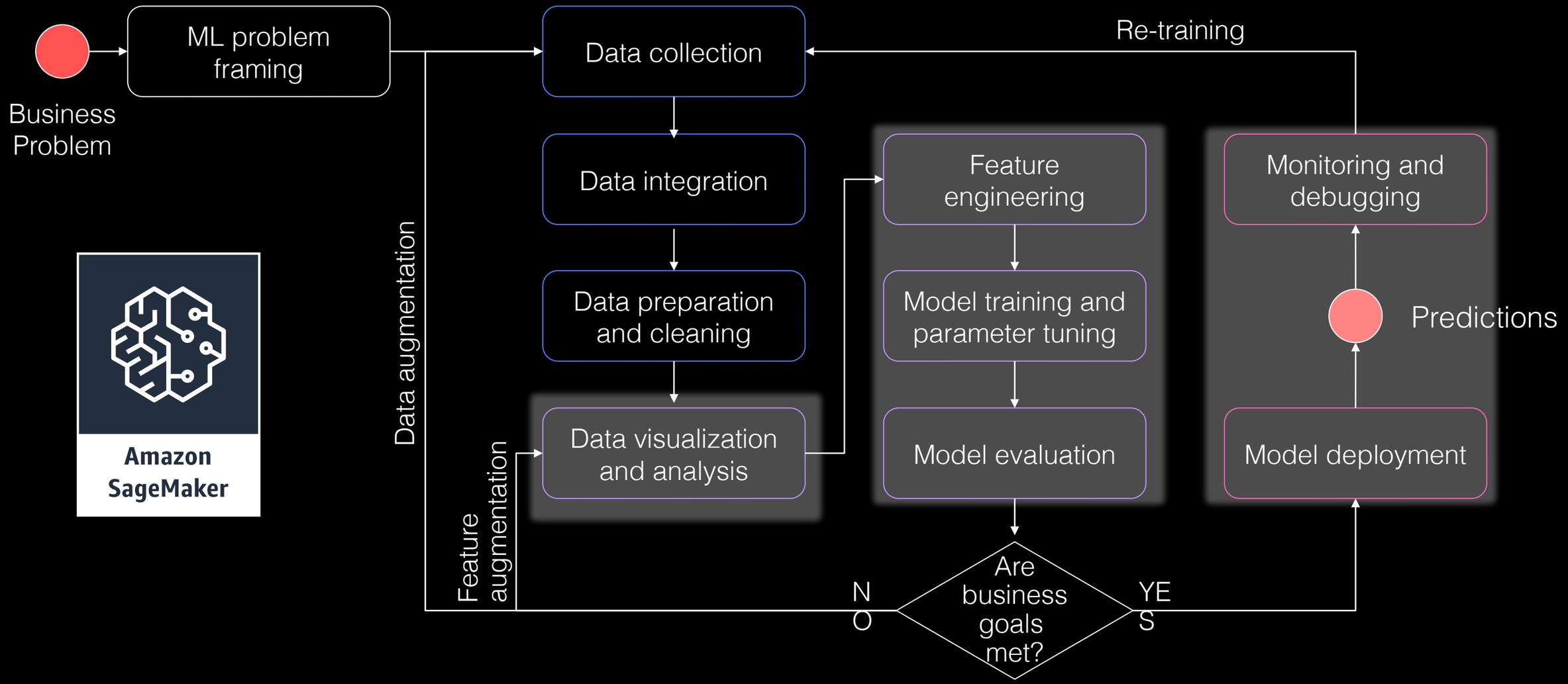
Deploy models in production



Scale and manage the production environment

Same service and APIs from experimentation to production

# Build, train and deploy models using SageMaker



# The Amazon SageMaker API

- Python SDK orchestrating all Amazon SageMaker activity
  - High-level objects for algorithm selection, training, deploying, automatic model tuning, etc.
  - Spark SDK (Python & Scala)

- AWS SDK
  - For scripting and automation
  - CLI : '*aws sagemaker*'
  - Language SDKs: boto3, etc.

# Model options

Training code

Factorization Machines
Linear Learner
Principal Component Analysis
K-Means Clustering
XGBoost
And more

Built-in Algorithms (17)

No ML coding required
No infrastructure work required
Distributed training
Pipe mode

mxnet Chainer TensorFlow PyTorch scikit-learn

Built-in Frameworks

Bring your own code: script mode
Open source containers
No infrastructure work required
Distributed training
Pipe mode

Bring Your Own Container

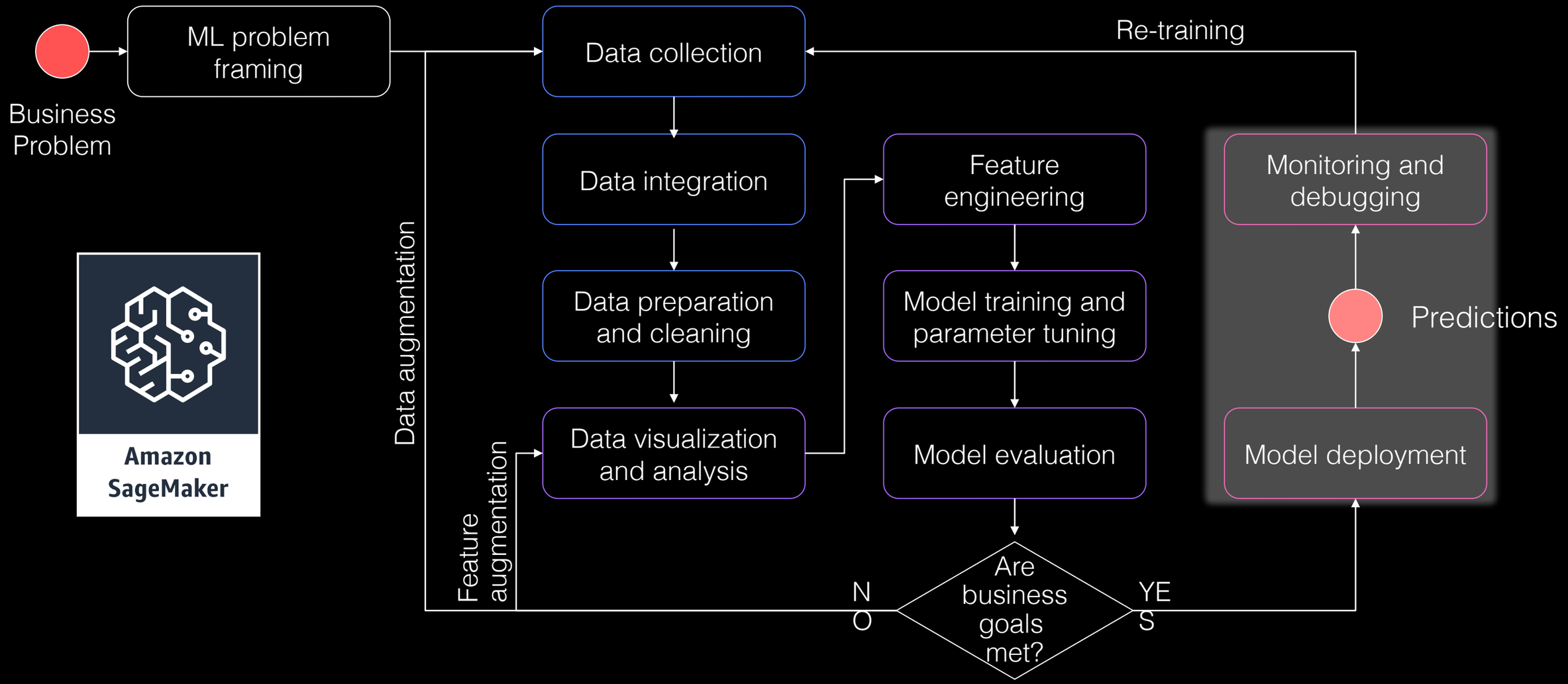
Full control, run anything!
R, C++, etc.
No infrastructure work required

# Built-in algorithms

orange: supervised, yellow: unsupervised

| | |
|---|---|
| **Linear Learner**: regression, classification | **Image Classification**: Deep Learning (ResNet) |
| **Factorization Machines**: regression, classification, recommendation | **Object Detection (SSD)**: Deep Learning (VGG, ResNet) |
| **K-Nearest Neighbors**: non-parametric regression and classification | **Neural Topic Model**: topic modeling |
| **XGBoost**: regression, classification, ranking https://github.com/dmlc/xgboost | **Latent Dirichlet Allocation**: topic modeling (mostly) |
| **K-Means**: clustering | **Blazing Text**: GPU-based Word2Vec, and text classification |
| **Principal Component Analysis**: dimensionality reduction | **Sequence to Sequence**: machine translation, speech to text |
| **Random Cut Forest**: anomaly detection | **DeepAR**: time-series forecasting (RNN) |
| **Object2Vec**: general-purpose embedding | **IP Insights**: usage patterns for IP addresses |
| **Semantic Segmentation**: Deep Learning | |

# Optimize and deploy models using SageMaker

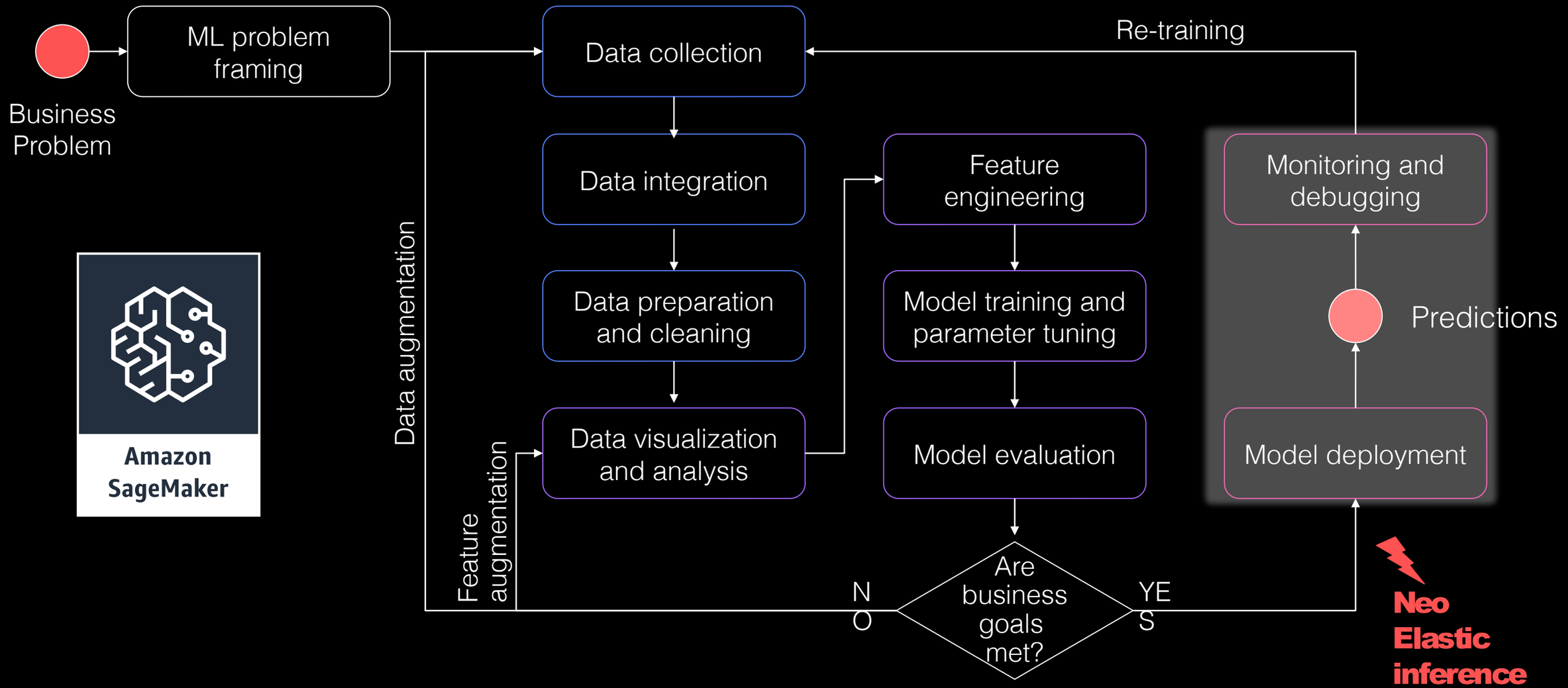


# Demo:

# Built-in image classification on Caltech-256

https://gitlab.com/juliensimon/dlnotebooks/blob/master/sagemaker/06-Image-classification-deeplens.ipynb

# Optimize and deploy models using SageMaker



# Amazon SageMaker Neo

Optimize models for the underlying hardware architecture

- Train once, run anywhere
- Supported frameworks and algorithms
  - TensorFlow, Apache MXNet, PyTorch, ONNX, and XGBoost
- Supported hardware architectures
  - ARM, Intel, and NVIDIA
  - More hardware architectures coming.

The Neo compiler and runtime are open source, enabling hardware vendors to customize it for their processors and devices: https://github.com/neo-ai/

# Compiling ResNet-50 for the Raspberry Pi

Configure the compilation job

```
{
  "RoleArn":$ROLE_ARN,
  "InputConfig": {
   "S3Uri":"s3://jsimon-neo/model.tar.gz",
   "DataInputConfig": "{\"data\": [1, 3, 224, 224]}",
   "Framework": "MXNET"
  },
 "OutputConfig": {
  "S3OutputLocation": "s3://jsimon-neo/",
  "TargetDevice": "rasp3b"
 },
 "StoppingCondition": {
 "MaxRuntimeInSeconds": 300
 }
}
```

Compile the model

```
$ aws sagemaker create-compilation-job
--cli-input-json file://config.json
--compilation-job-name resnet50-mxnet-pi

$ aws s3 cp s3://jsimon-neo/model-
rasp3b.tar.gz .

$ gtar tfz model-rasp3b.tar.gz
compiled.params
compiled_model.json
compiled.so
```

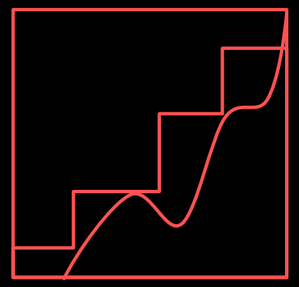
Predict with the compiled model

```
from dlr import DLRModel
model = DLRModel('resnet50', input_shape,
output_shape, device)
out = model.run(input_data)
```

# Amazon Elastic Inference

## Attach fractional acceleration to any EC2 instance

Lower inference costs
up to 75%



Match capacity
to demand

Available between 1 to 32
TFLOPS

Integrated with
Amazon EC2,
Amazon SageMaker,
and Amazon DL
AMIs

Support for TensorFlow,
Apache MXNet, and
ONNX
with PyTorch coming soon

Single and
mixed-precision
operations

# Demo:

# Image classification with Keras and Elastic Inference

https://gitlab.com/juliensimon/dlnotebooks/tree/master/keras/05-keras-blog-post

# Getting started

http://aws.amazon.com/free

https://ml.aws
https://aws.amazon.com/sagemaker

https://github.com/aws/sagemaker-python-sdk
https://github.com/aws/sagemaker-spark
https://github.com/awslabs/amazon-sagemaker-examples
https://gitlab.com/juliensimon/ent321

https://medium.com/@julsimon
https://gitlab.com/juliensimon/dlnotebooks



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Merci!

Julien Simon
Global Evangelist, AI & Machine Learning, AWS
@julsimon