Pragmatic machine learning for business

Jakub Nowacki

Confitura 2018

whoami

Lead Data Scientist @ SigDelta (sigdelta.com)

Trainer @ Sages (sages.com.pl)

I can code, I do maths

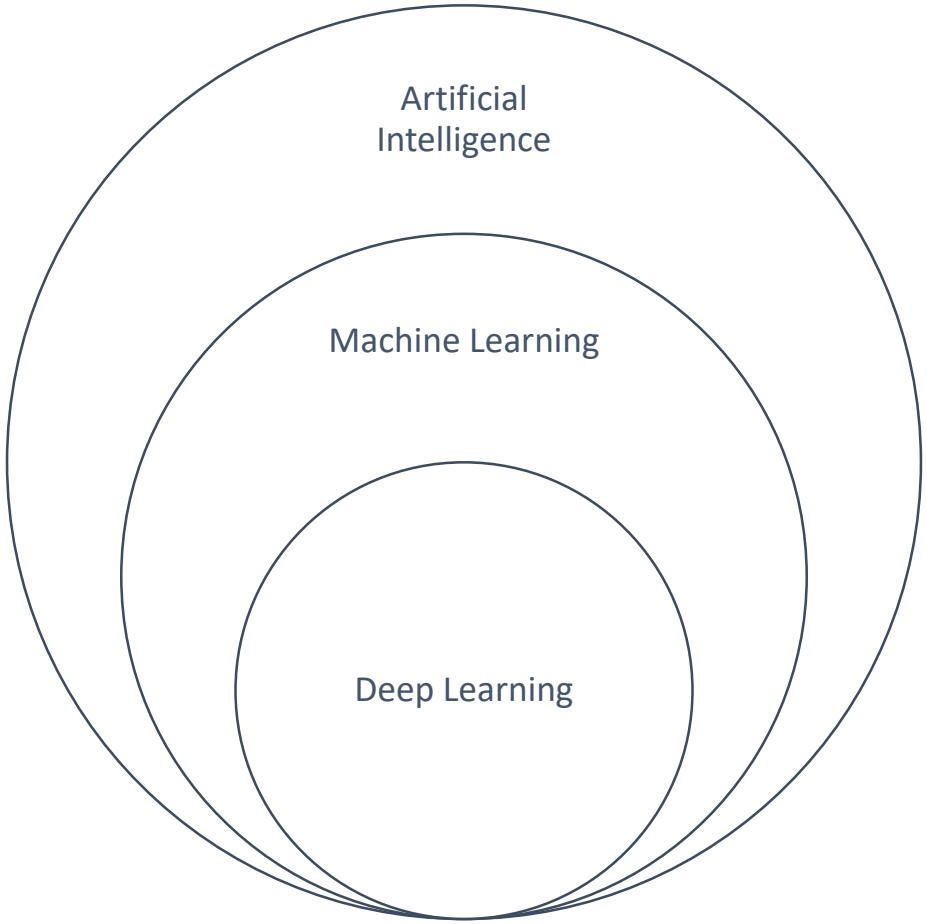
@jsnowacki

What is Machine Learning?

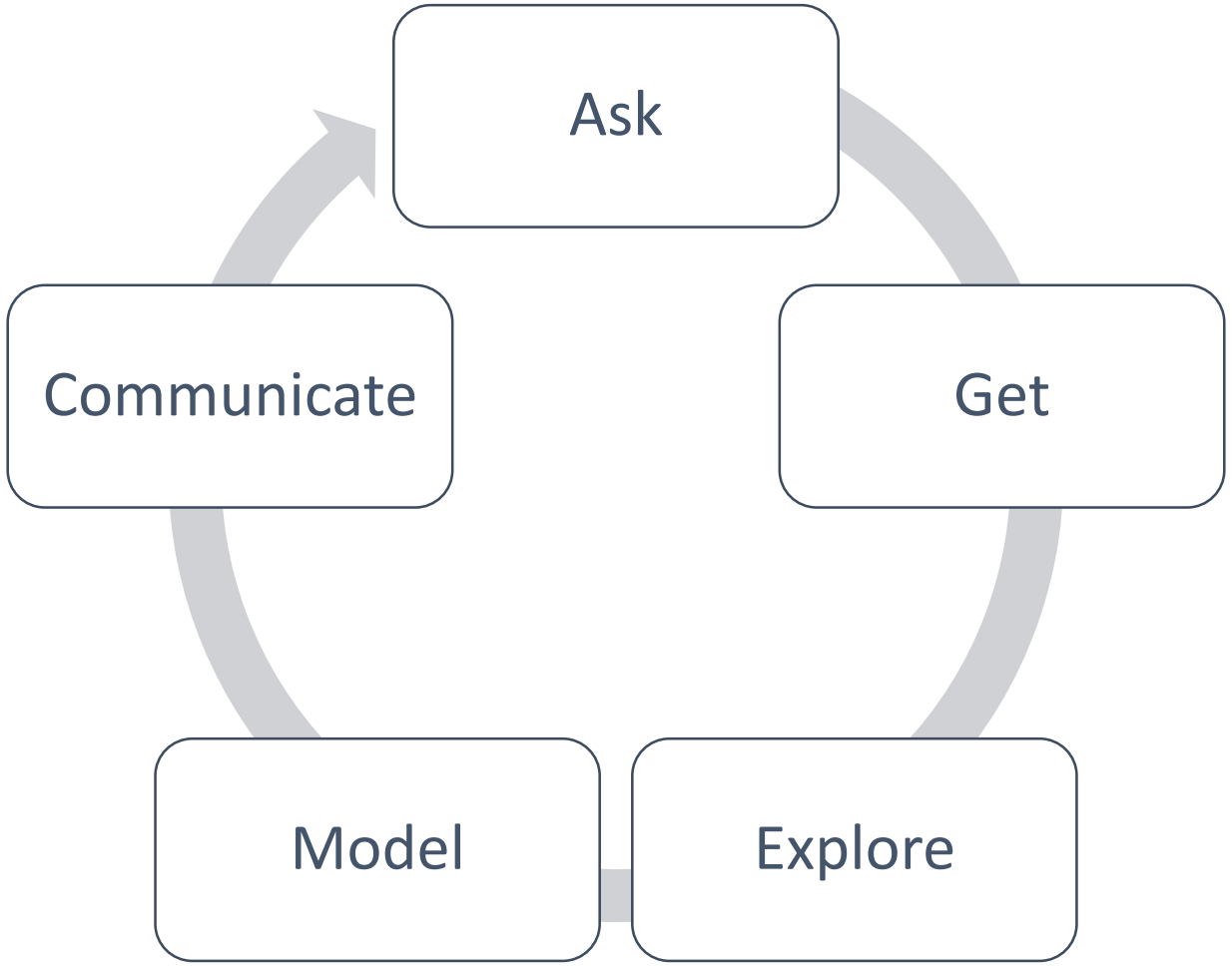
Machine learning is a subset of artificial intelligence in the field of computer science that often uses statistical techniques to give computers the ability to "learn" with data, ***without being explicitly programmed****.*

Wikipedia, https://en.wikipedia.org/wiki/Machine_learning

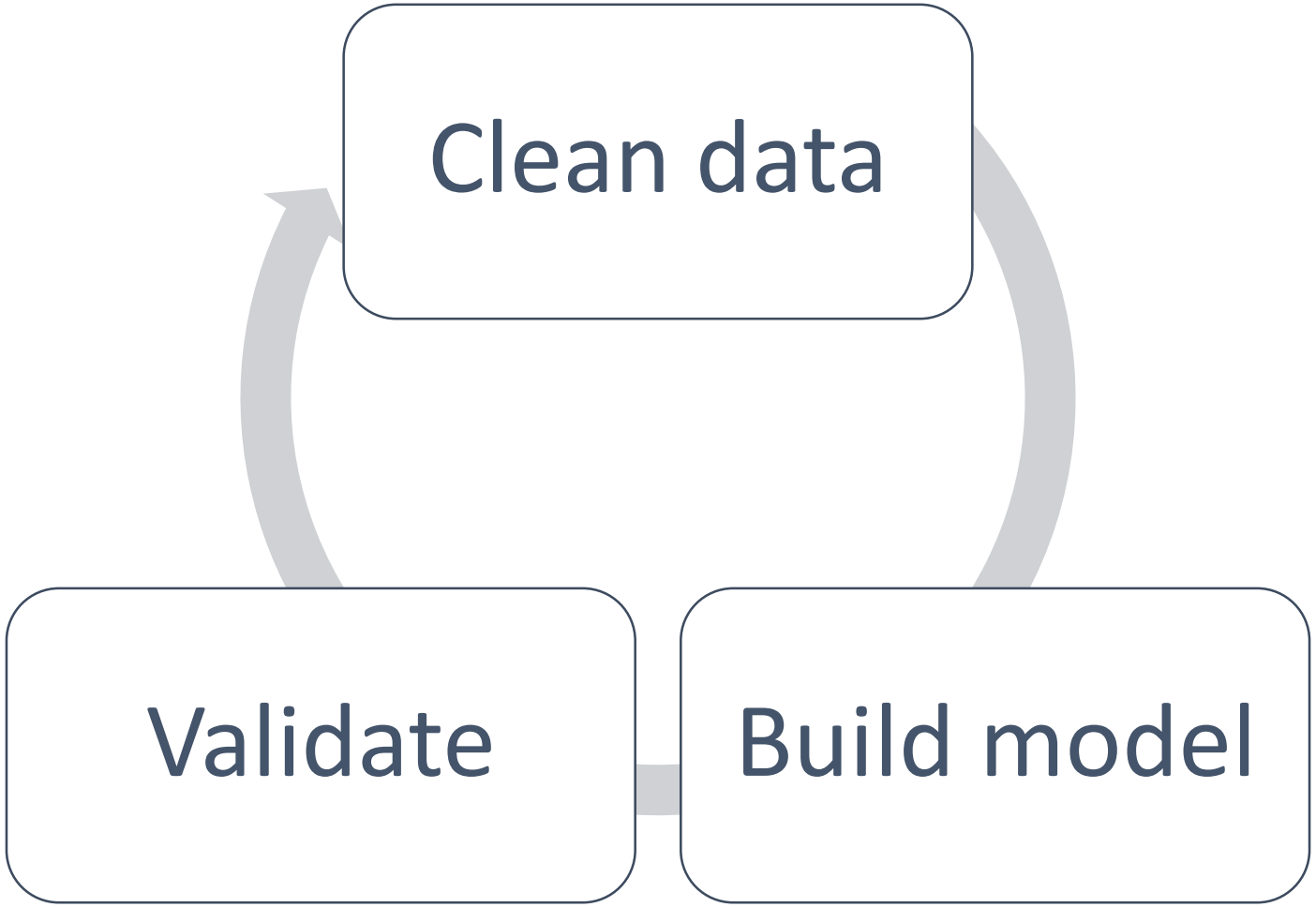
ML vs AI



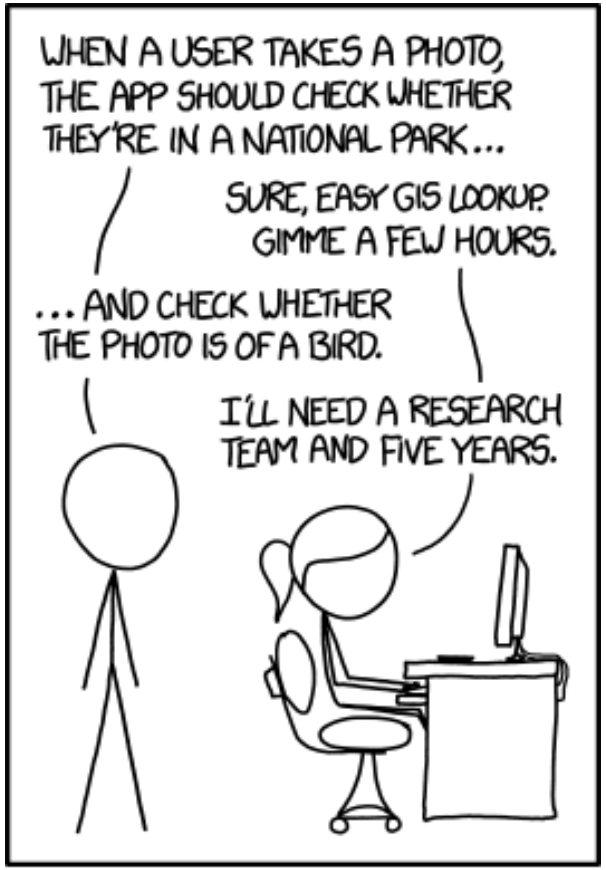
Data Science process



How Machine Learning usually works?



Is it hard?



Source: https://xkcd.com/1425/

AI APIs

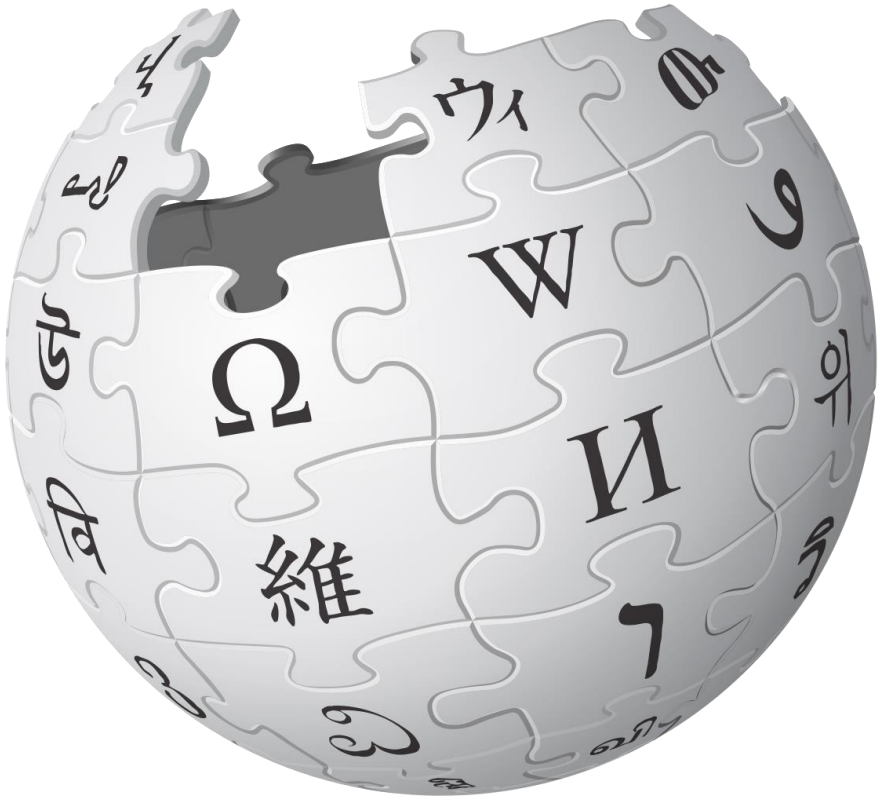
Source: https://aws.amazon.com/rekognition/

Available models

Source: https://dev.to/swyx/serverless-machine-learning-at-google-cp9

Available data

Source: https://en.wikipedia.org/wiki/MNIST_database



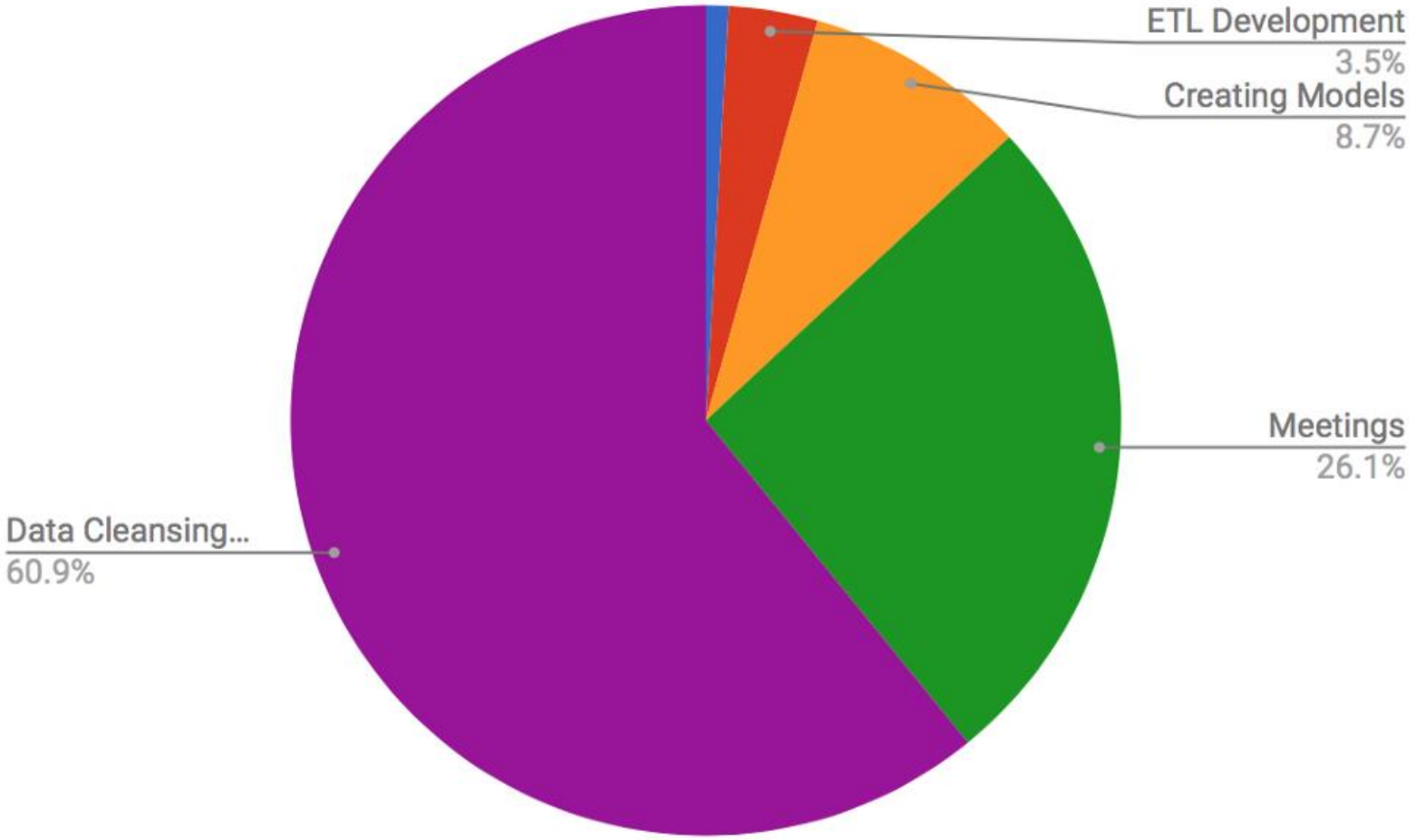
https://dumps.wikimedia.org/

What should we do now?

- Data cleaning
- Data annotation
- Model training
- Transfer learning
- Model deployment

Source: https://www.coursera.org/learn/machine-learning

Data cleaning



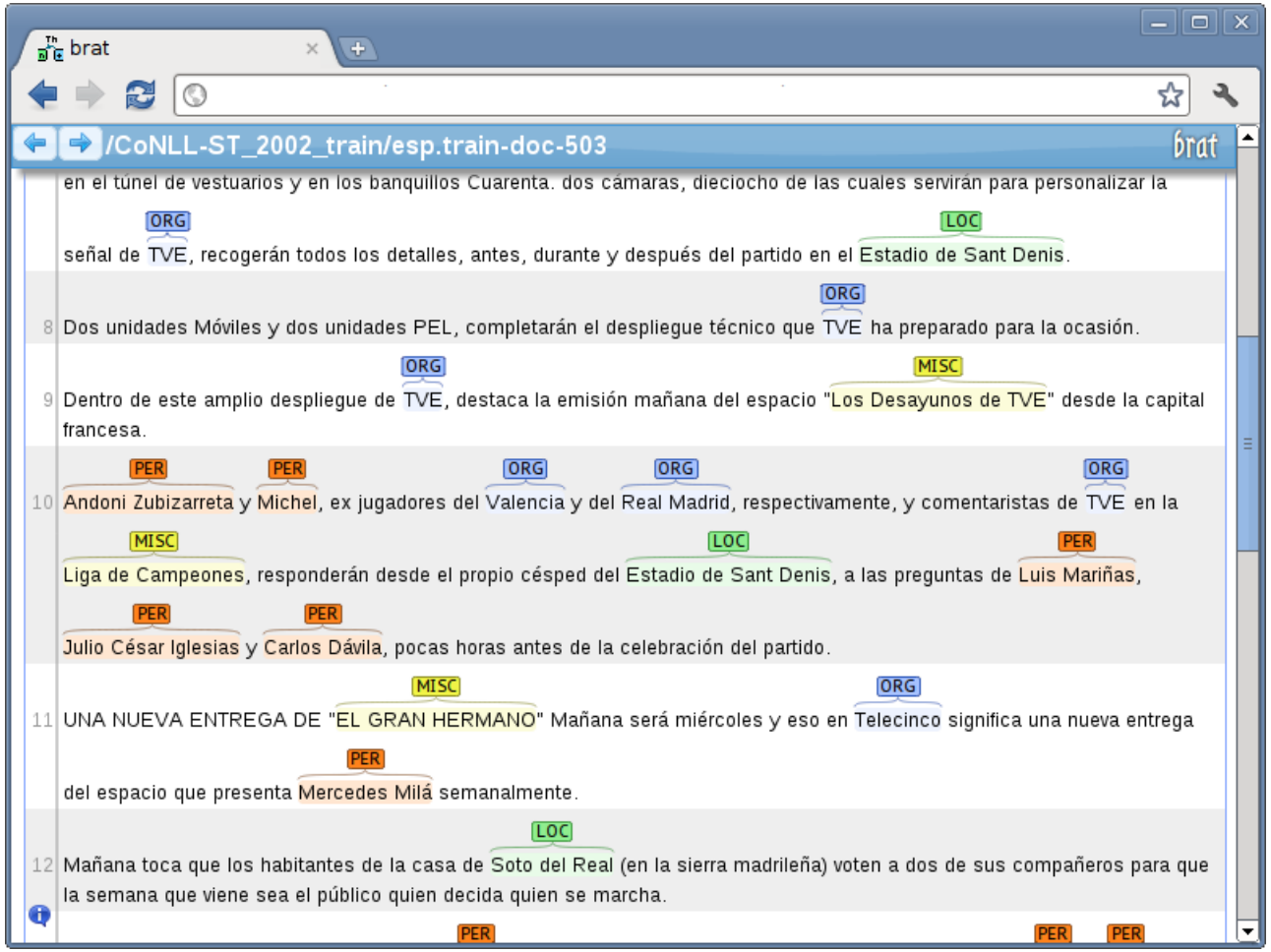
Source: https://towardsdatascience.com/intro-to-data-analysis-for-everyone-part-3-d8f02690fba0

Data cleaning - TensorFlow Data

```
dataset2 = tf.data.Dataset.from_tensor_slices(
       (tf.random_uniform([4]),
        tf.random_uniform([4, 100], maxval=100, dtype=tf.int32)))
print(dataset2.output_types) # ==> "(tf.float32, tf.int32)"
print(dataset2.output_shapes) # ==> "((), (100,))"

dataset3 = tf.data.Dataset.zip((dataset1, dataset2))
print(dataset3.output_types) # ==> (tf.float32, (tf.float32, tf.int32))
print(dataset3.output_shapes) # ==> "(10, ((), (100,)))"

dataset1 = dataset1.map(lambda x: ...)
dataset2 = dataset2.flat_map(lambda x, y: ...)
dataset3 = dataset3.filter(lambda x, (y, z): ...)
```

Source: https://www.tensorflow.org/programmers_guide/datasets

Data annotatnion



Source: http://brat.nlplab.org/

Custom model

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
optimizer=keras.optimizers.Adadelta(), metrics=['accuracy'])

model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, verbose=1,
validation_data=(x_test, y_test))
```

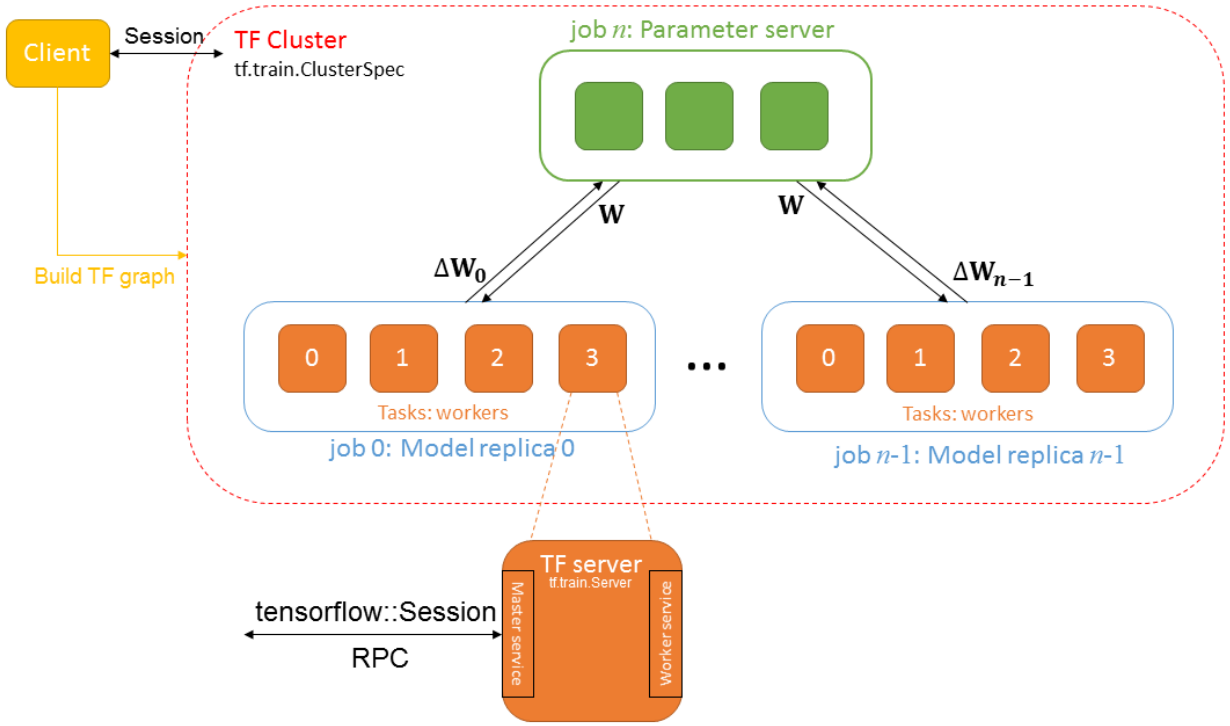
MINST accuracy: 99.25%

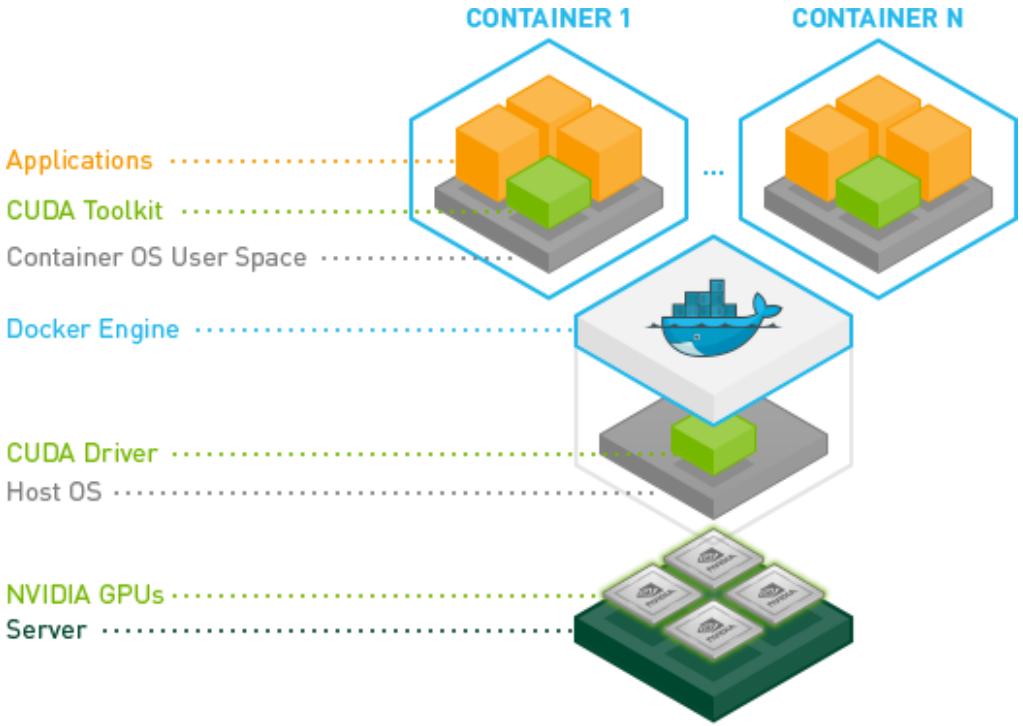
Custom model

“Finishing a 90-epoch ImageNet-1k training with ResNet-50 on a NVIDIA M40 GPU takes 14 days.”

Yang You et al., *ImageNet Training in Minutes*, 2018

Custom model - TensorFlow Distributed

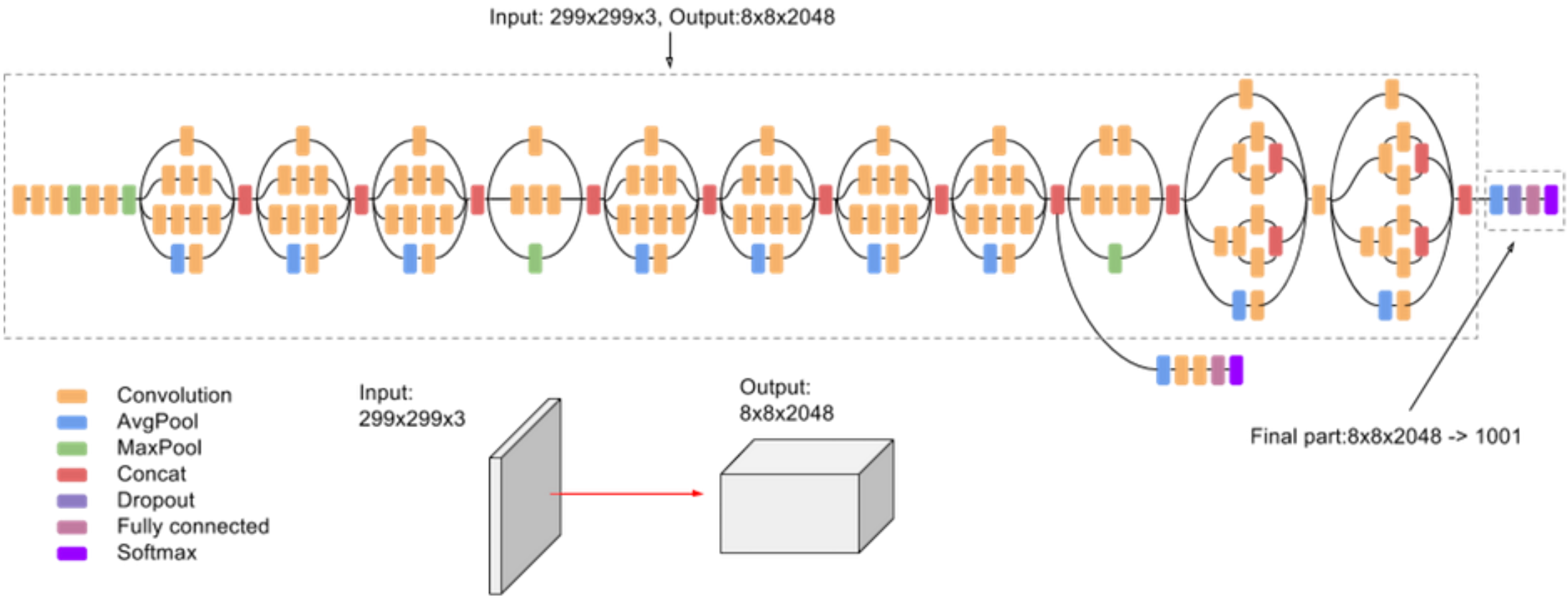




Source: http://www.pittnuts.com/2016/08/glossary-in-distributed-tensorflow/

Source: https://towardsdatascience.com/using-docker-to-set-up-a-deep-learning-environment-on-aws-6af37a78c551

Transfer learning



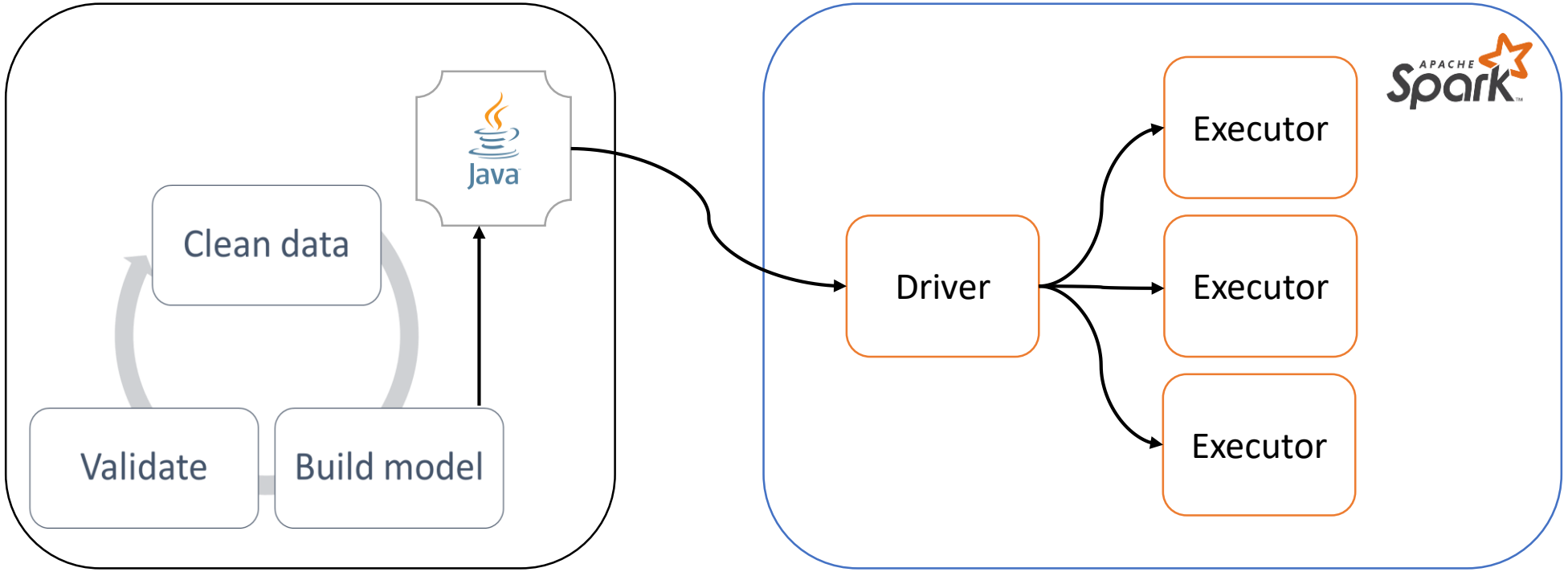
Source: https://cloud.google.com/tpu/docs/inception-v3-advanced

Transfer learning – TensorFlow Estimator & Hub

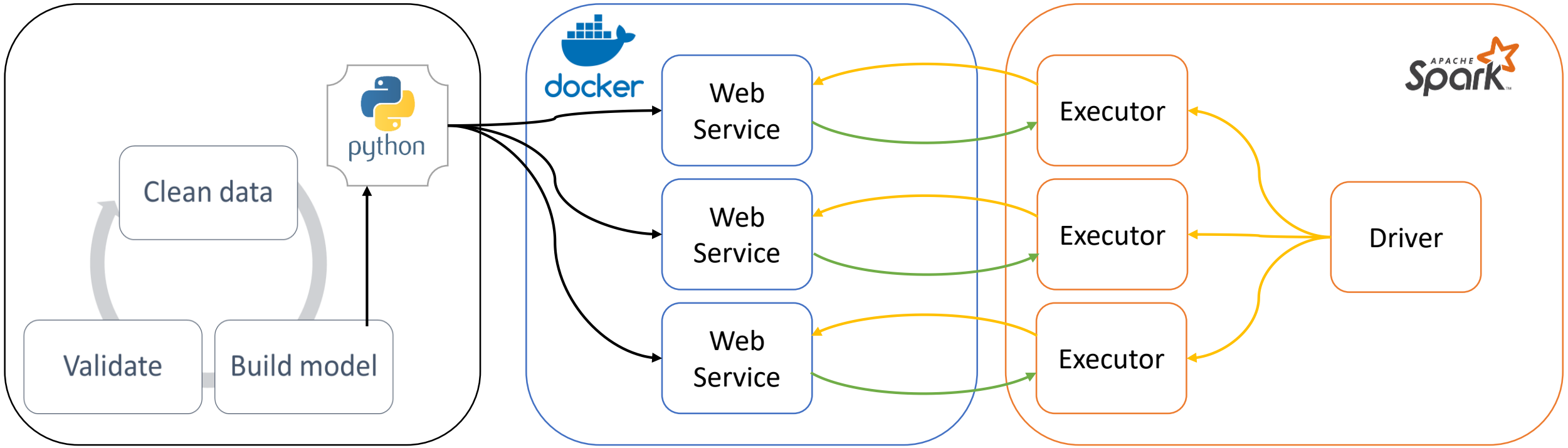
```
…
module =
hub.Module("https://tfhub.dev/google/imagenet/inception_v3/feature_vector/1")
input_layer = adjust_image(features["x"])
outputs = module(input_layer)
logits = tf.layers.dense(inputs=outputs, units=10)

predictions = {
  "classes": tf.argmax(input=logits, axis=1),
  "probabilities": tf.nn.softmax(logits, name="softmax_tensor")
}

if mode == tf.estimator.ModeKeys.PREDICT:
      return tf.estimator.EstimatorSpec(mode=mode, predictions=predictions)
…
```

Source: https://github.com/shu-yusa/tensorflow-hub-sample/blob/master/inceptionv3.py

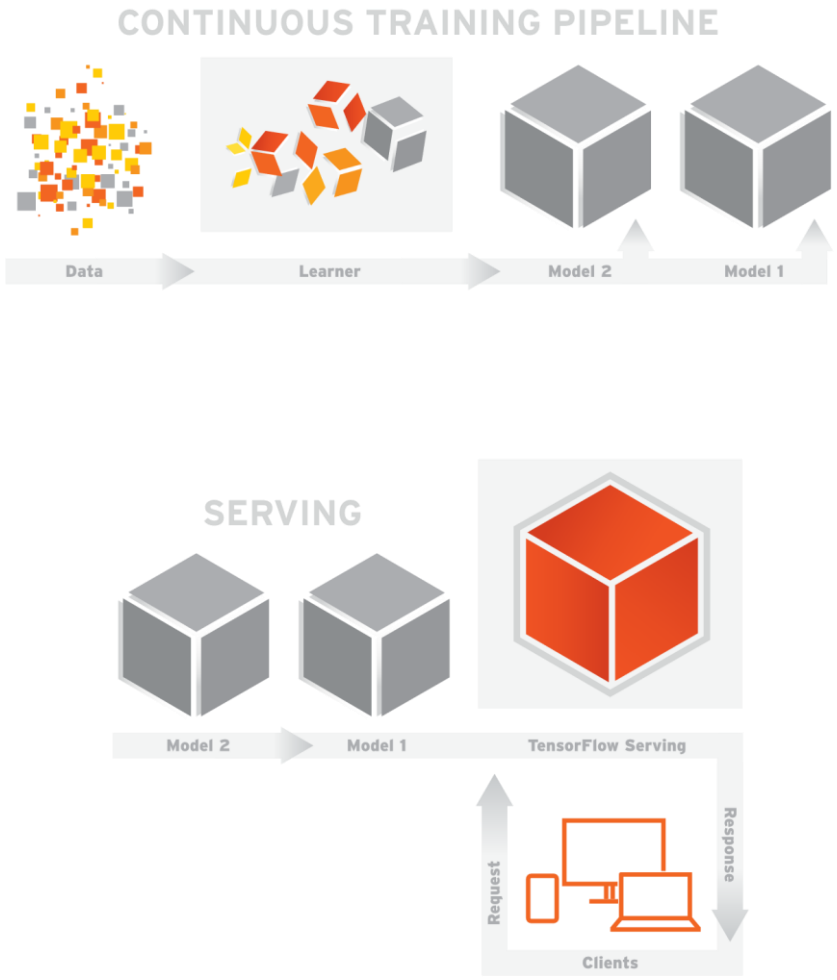
Deployment – model embedding



Deployment – model in containers



Deployment - TensorFlow Serving



Source: https://www.tensorflow.org/serving/

Source: https://cloud.google.com/products/machine-learning/

Thank you!

Questions?