Eto

Rikai: analytics on unstructured data at scale

Chang She and Lei Xu

DataCouncil 2022-03-24

Who we are

Eto

Chang She

@changhiskhan (Twitter)

Pandas (2nd core contributor)

VP Eng @ Tubi TV

CTO Datapad (acq. By Cloudera)

Lei Xu

ML Platform Lead @ Cruise

Hadoop PMC @ Cloudera

PhD Distributed Systems

Eto

Data management for ML datasets suck

We need a better way to organize, access, and analyze images, videos, text, and other unstructured data

Introducing Rikai (理解):

Eto

A framework designed for ML workflows focused on managing, exploring, and analyzing unstructured datasets (github.com/eto-ai/rikai)

Data format

- Rich semantic types on top of parquet
- Explicit abstractions for images, videos, text (future), geo, sensors
- Explicit types for bounding boxes, labels, and other annotations

SQL-ML extensions

- Train, eval, infer custom pytorch/tensorflow models using SQL
- Bring standard analytics tooling to ML datasets
- ML specific UDFs for areas, IOUs, crops, similarity, and more

IO connectors

- Low-code ingestion from popular open-source data formats
- PyTorch/Tensorflow/pandas/spark
- Integrates with dbt / BI tooling via JDBC

Rikai Data Format

Eto

- Apache Parquet + Semantic Types
- Pandas/Torch/Tf readers
- Scan from Presto/Redshift/BigQuery
- Spark UDT for SerDe (for now)

```
class Image(ToNumpy, Asset, Displayable):
  __UDT__ = ImageType()

class ImageType(UserDefinedType):
  @classmethod
  def sqlType(cls) -> StructType:
    return StructType(
      fields=[
        StructField("data", BinaryType(),
nullable=True),
        StructField("uri", StringType(), nullable=True),
      ])
```

Semantic Types

Eto

- Rich set of ML domain-specific types (images, video, annotations, etc)
- Useful helper methods and mixins
  - Image: to / from { PIL, Numpy, Tensors }
  - Bounding Box: Chop , IOU, Area Size.
  - Numpy / Tensor interop
  - Jupyter display
- Rikai knows when and how to convert parquet data to {torch/tf}.Tensor
  - **Box2d(x1, y1, x2, y2)** => **torch.Tensor([x1, y1, x2, y2], device=gpu)**
  - **Image(uri="s3://.../foo.png") => torch.Tensor([....], shape=(32, 32, 3), device=gpu)**

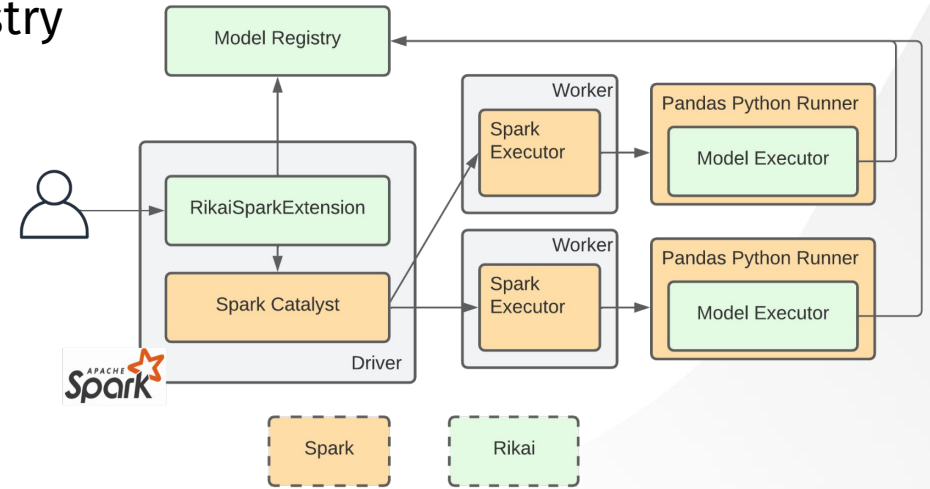
Example: Use Rikai in PyTorch

Eto

```
from rikai.torch.data import Dataset
dataset = Dataset("s3://bucket/path/to/dataset")
# Compatible with pytorch DataLoader
loader = torch.utils.data.DataLoader(dataset, batch_size=8, num_workers=8)
model.eval()
for batch in loader:
   print(batch)
   predictions = model(batch)
# Sample output:
# {'mask': tensor([[[0.9037, 0.9284, 0.6832, 0.5378], ..., dtype=torch.float64),
#  'id': tensor([997]),
#  'image': tensor([[[  5,   7,  52,  ...,  35,  74,  16],
#  [ 25,  62,  91,  ..., 114,  71,  27]]], dtype=torch.uint8)},
```

SQL-ML

- Extends Spark SQL via [SparkSessionExtensions]
  - **ML_PREDICT(model, col, ...)**
  - **ML_EVAL(model, ground_truth, col)** *
- JIT + Pandas UDF as Model Executor
  - Using Model Metadata in Registry
  - GPU batch inference
- Pytorch, Tensorflow, Sklearn



Analyze ML dataset via SQL

Eto

```
1   -- Find mislabeled data
2   SELECT image_id, image FROM (
3       SELECT
4           image_id,
5           image,
6           ground_truth,
7           ML_PREDICT(yolov5, image) as yolo,
8           ML_PREDICT(ssd, image) as ssd
9       FROM coco WHERE split = "eval"
10  ) WHERE iou(ssd, yolo) - iou(yolo, ground_truth) > THRESHOLD
```

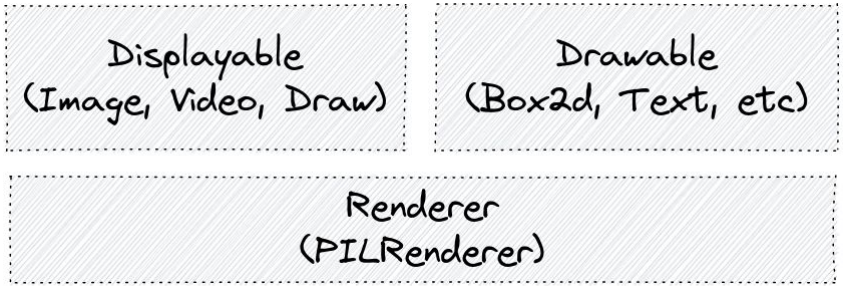
IO connectors

- Frameworks:
  - Spark: **df.write.format(“rikai”).save()**
  - PyTorch: **rikai.torch.data.Dataset**
  - Tensorflow: **rikai.tensorflow.data.from_rikai()**
- Data formats:
  - COCO: computer vision
  - ROS Bag: Robot OS format
  - NuScenes*, Kitti*, and more

Visualizing unstructured data

Eto

- Need to view images, annotations, labels, scores
- Tooling is too low level - often need OpenCV / PIL APIs directly
- But Rikai semantic types enable us to build a higher level API




```
class ImageDraw(Draw):
    def __init__(self, img: Image):
        super().__init__()
        self.img = img.to_pil()

    def to_image(self) -> Image:
        if not self.layers:
            raise ValueError("Can not render

        render = PILRenderer(self.img)
        for layer in self.layers:
            layer._render(render)
        return Image.from_pil(render.image)
```

Visualizing unstructured data - a quick example

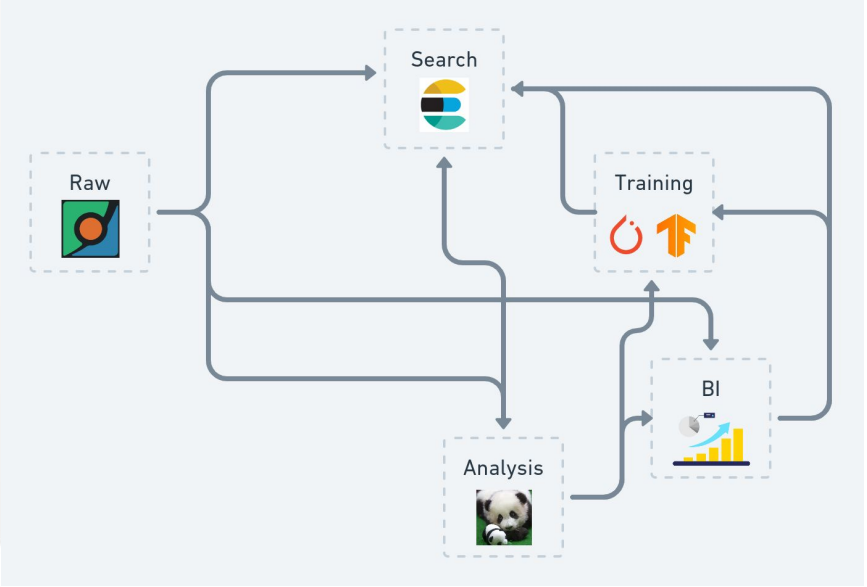
```
for rect, label in zip(bboxes, labels):
    x,y,w,h = rect
    cv2.rectangle(image,
                  (x, y),
                  (x+w, y+h),
                  (0, 0, 255), 2)
cv2.imwrite('two_blobs_result.jpg',result)
cv2.imshow("bounding_box", result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

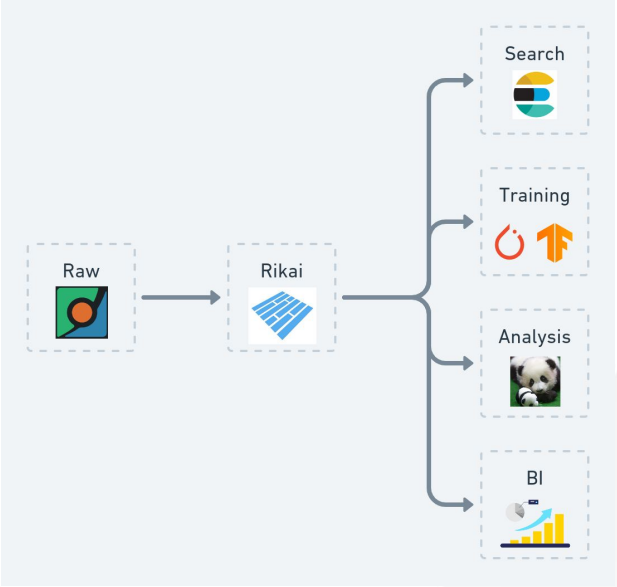


```
image | bboxes@{'color': 'blue', 'width': 2}
```

Obligatory before and after slide

Eto





- Sane data management abstractions
- Single-source of truth
- Bring standard data tooling to ML datasets

Rikai roadmap

Eto

Core format

- Text, Sensor, Geo
- More annotations
- Native writer

SQL extension

- Better UDT access
- More native UDFs
- Presto / Dask
- Performance tuning

Model support

- Out-of-box models
- ONNX format
- TensorRT
- Train / eval syntax
- Model catalog

Visualization

- Auto label position
- In-notebook curation
- Search / filter controls

Eto

Thank You.

We'd love to hear your feedback, bug reports, and feature requests on Github: https://github.com/eto-ai/rikai

Rikai contributors (rikai-dev@eto.ai)

- Da Shen
- Renkai Ge
- Chuck Yang
- Wenbing Bai
- Feifei Cai
- Terry Rodriguez