Building Efficient ML Pipelines and Responsible AI Solutions

Adi Polak

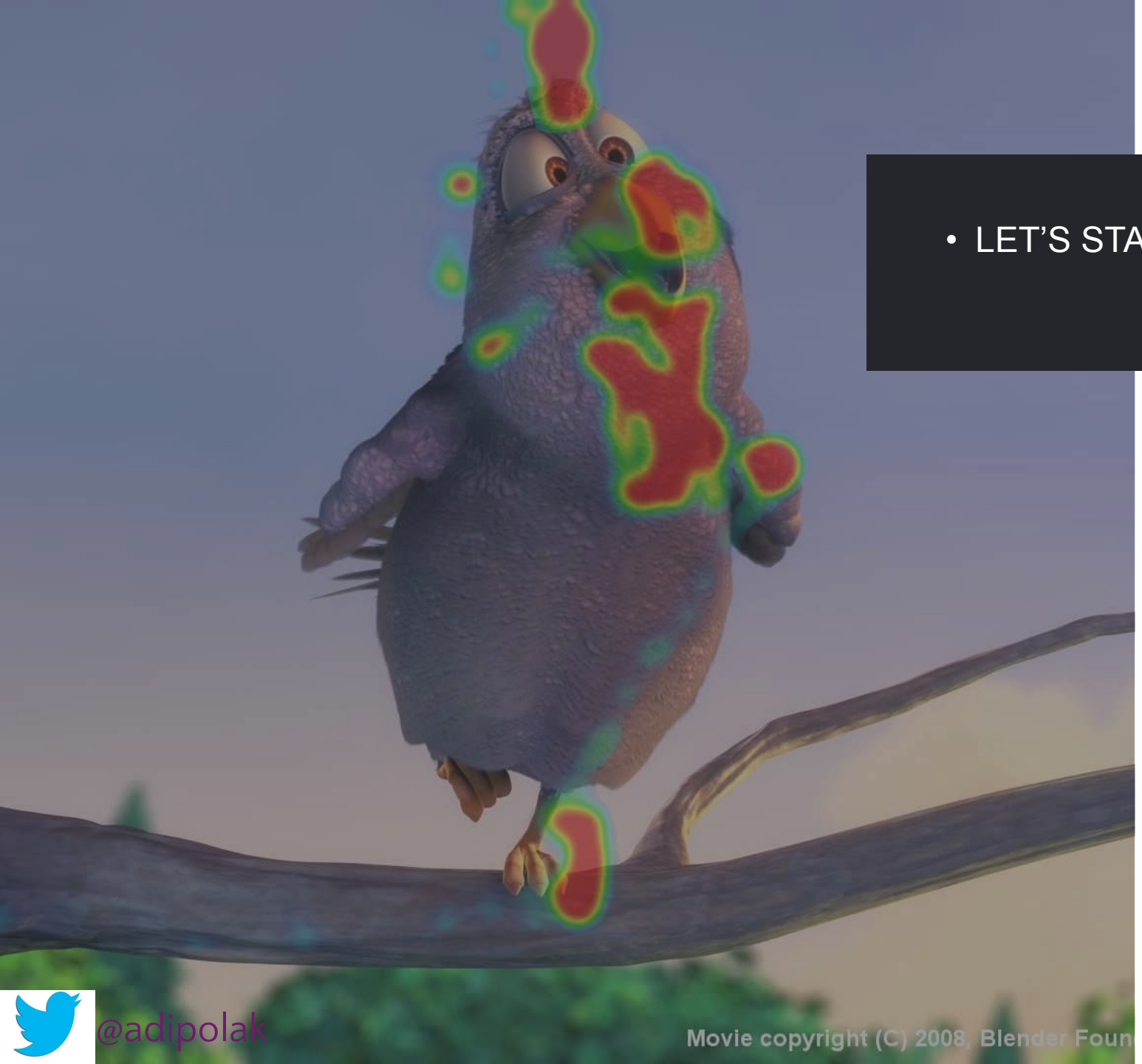
Microsoft

@adipolak


CODE
WATER

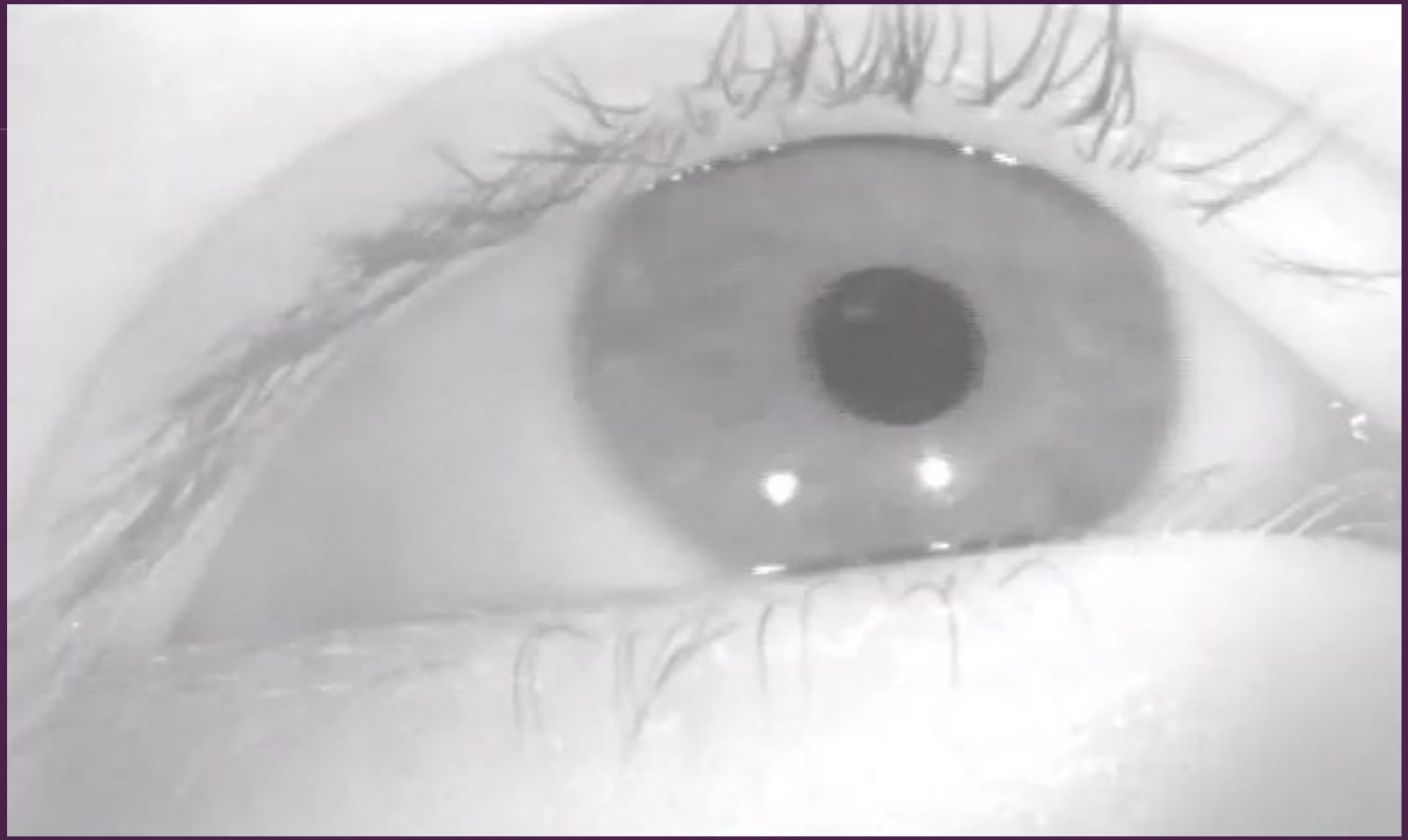

COMMUNITY

Trust



- LET'S START FROM THE BEGINNING.

What happens when we get raw data?



@adipolak

ML Process / Life Cycle

1. Gather Data
2. Feature Extract, Clean and Normalize
3. Select algorithm
4. Evaluate model
5. Data/Insights visualization

@adipolak

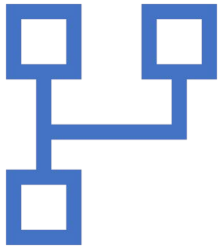
But in real life:

Accuracy < 0.5
ROC curve ☹️

Aim for high Accuracy

What can you do?

Automate!

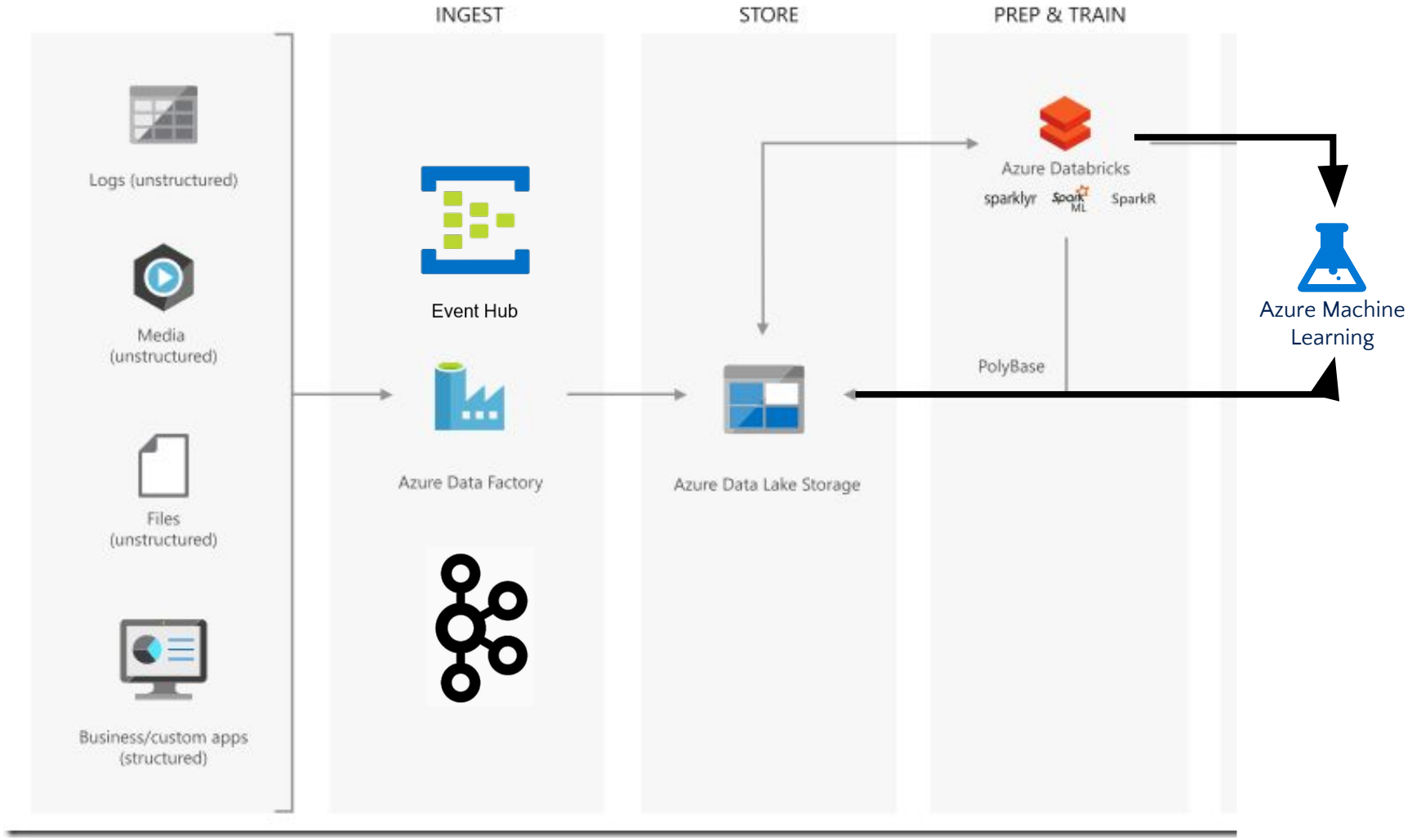


HOW?

Pipelines!

What are pipelines?

Big Data/ ML Pipelines



Demo

Apache Spark ML Pipelines

Configure ML pipeline stages

```
1
2  // Configure an ML pipeline, which consists of three stages: tokenizer, hashingTF, and lr.
3  val tokenizer = new Tokenizer()
4    .setInputCol("text")
5    .setOutputCol("words")
6
7
8  val hashingTF = new HashingTF()
9    .setNumFeatures(1000)
10   .setInputCol(tokenizer.getOutputCol)
11   .setOutputCol("features")
12
13
14 val lr = new LogisticRegression()
15   .setMaxIter(10)
16   .setRegParam(0.001)
17
18 run.logParam("lr_maxInt", "10")
19 run.logParam("lr_RegParam", "0.001")
20
```

Show result

Set the pipeline

```
1  val pipeline = new Pipeline()
2    .setStages(Array(tokenizer, hashingTF, lr))
```

Show result

Fit the data

```
1 // Fit the pipeline to training documents.
2 val model : PipelineModel = pipeline.fit(training)
```

Save and load model

```
// Now we can optionally save the fitted pipeline to disk
model.write.overwrite().save("/tmp/spark-logistic-regression-model")

// We can also save this unfit pipeline to disk
pipeline.write.overwrite().save("/tmp/unfit-lr-model")

// And load it back in during production
val sameModel = PipelineModel.load("/tmp/spark-logistic-regression-model")
```

Create Mock test set

```
1  // Prepare test documents, which are unlabeled (id, text) tuples.
2  val test = spark.createDataFrame(Seq(
3    (4L, "spark i j k"),
4    (5L, "l m n"),
5    (6L, "spark hadoop spark"),
6    (7L, "apache hadoop")
7  )).toDF("id", "text")
8
```

Prepare file printer for MLflow atrifact file

```
1 val filePrinter = new PrintWriter("/tmp/output.txt")
2 filePrinter.write("LR run")
```

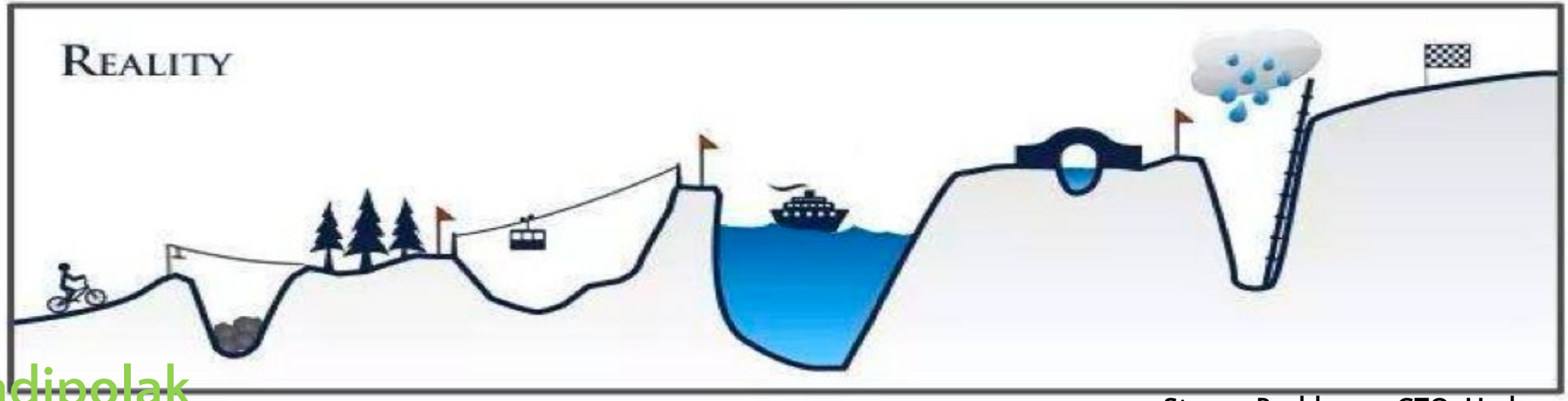
Make prediction using the ML model

```
// Make predictions on test documents.
model.transform(test)
  .select("id", "text", "probability", "prediction")
  .collect()
  .foreach { case Row(id: Long, text: String, prob: Vector, prediction: Double) =>
    //println(s"($id, $text) --> prob=$prob, prediction=$prediction")
    filePrinter.write(s"($id, $text) --> prob=$prob, prediction=$prediction")
  }
```

Close and save

```
1  run.logArtifact(Paths.get("/tmp/output.txt"))
2  run.endRun()
3  filePrinter.close()
```

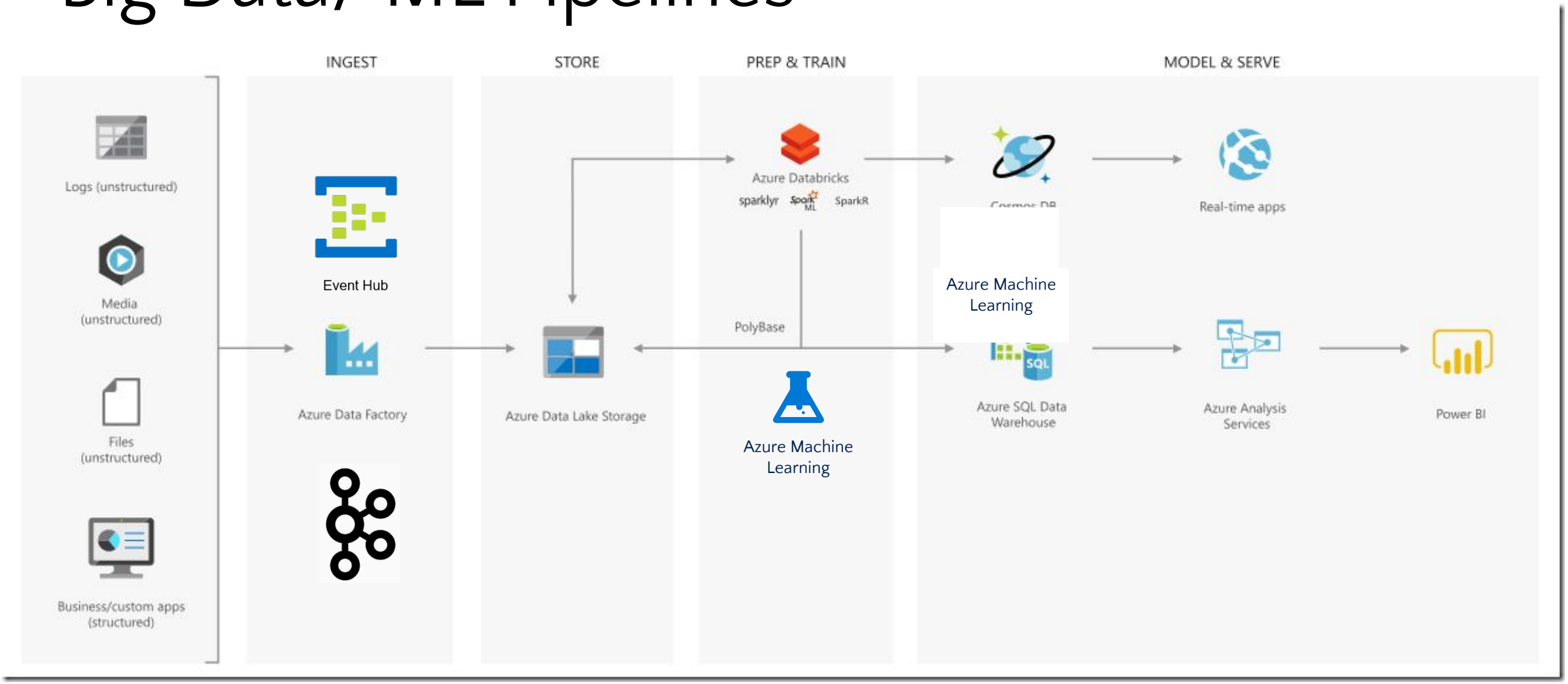
Show result



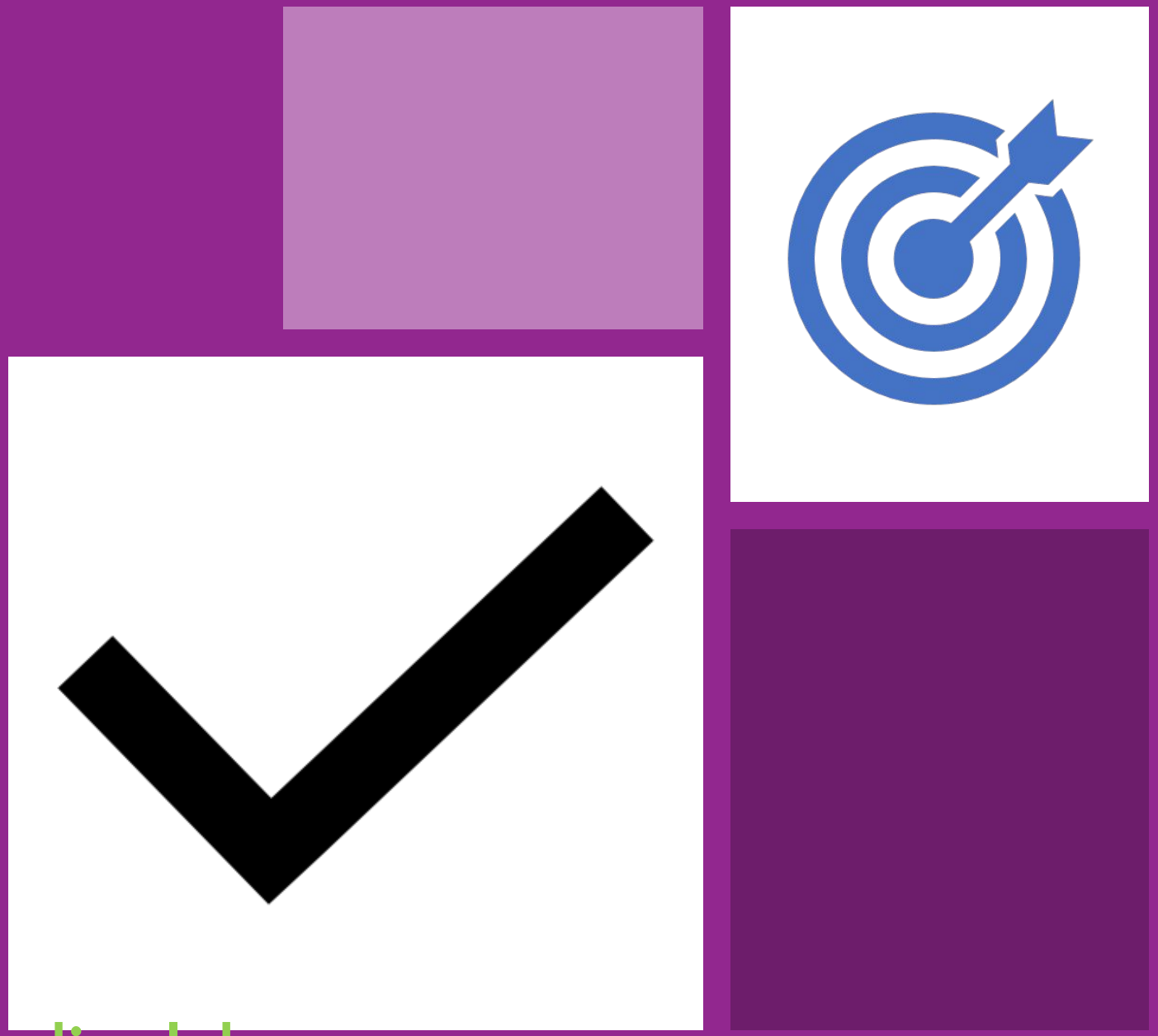
@adipolak

Stepan Pushkarev, CTO, Hydrosphere.i

Big Data/ ML Pipelines



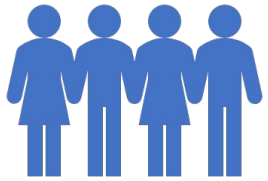
@adipolak



High accuracy!

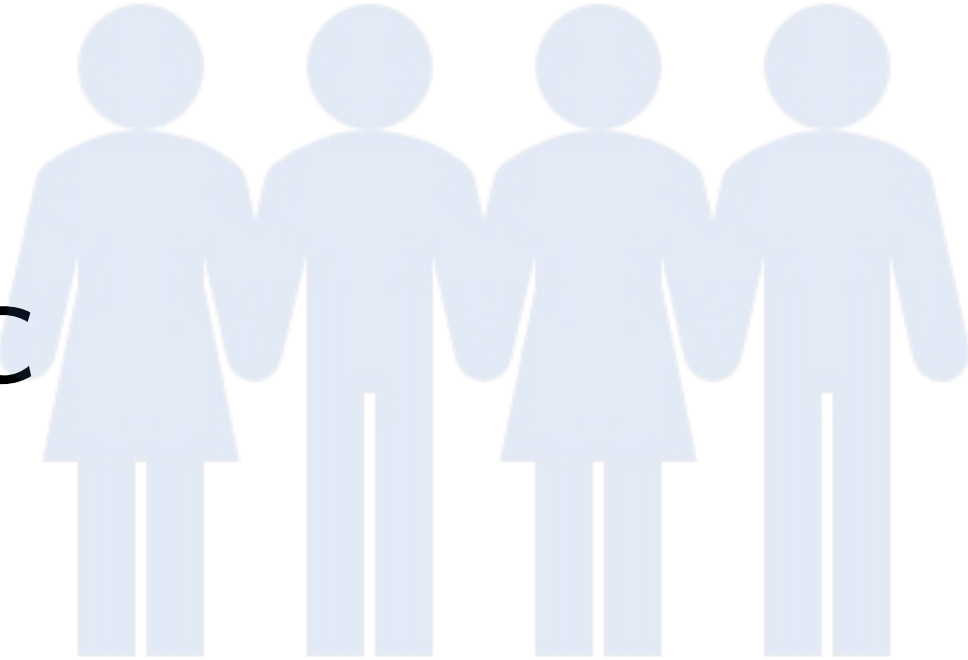
But, at what cost?

false positives



Human centric

Responsible AI



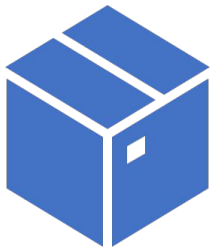

Pixabay

Our updated goals:

Lawful

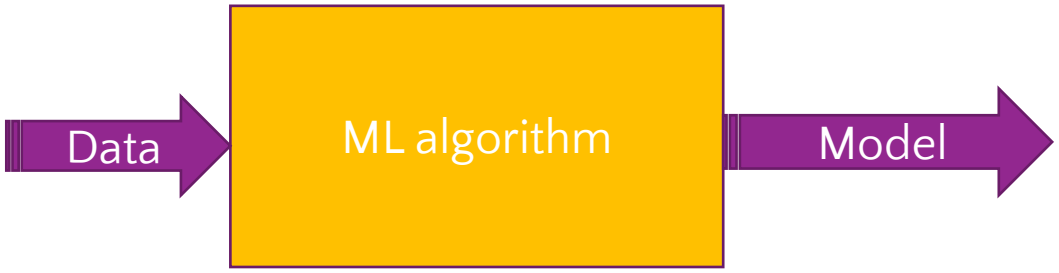
Ethical

Robust

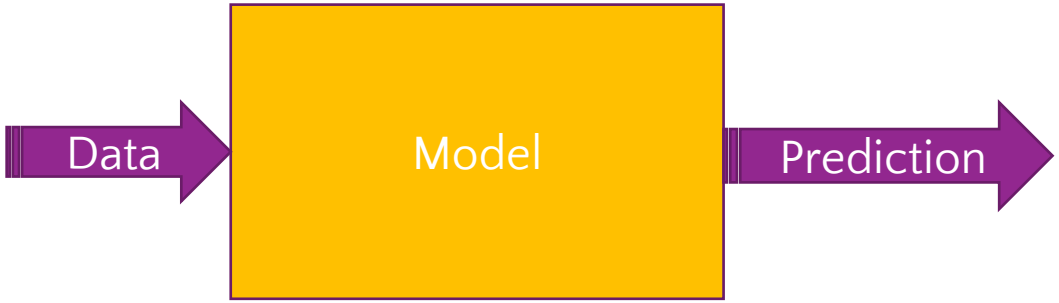


ML is a black box

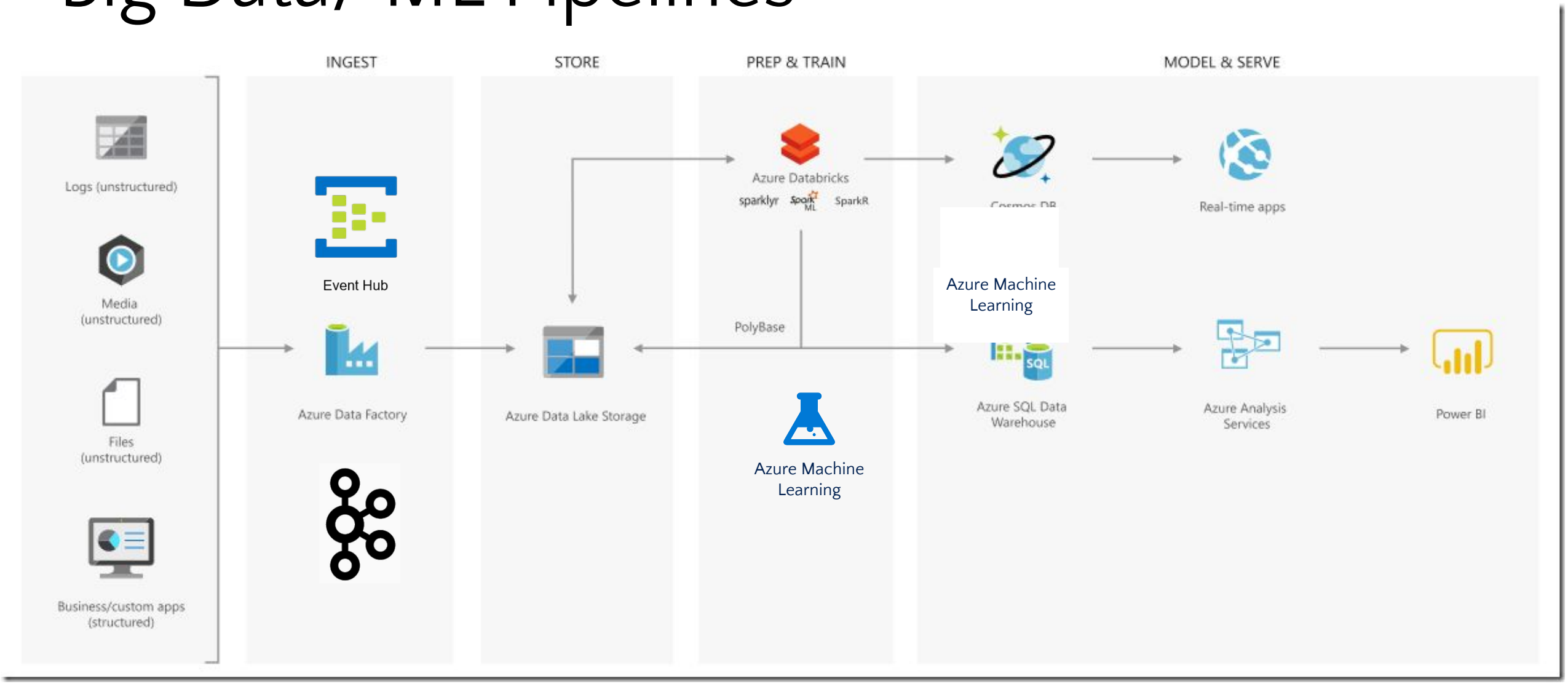
Training:



Testing/Prediction:

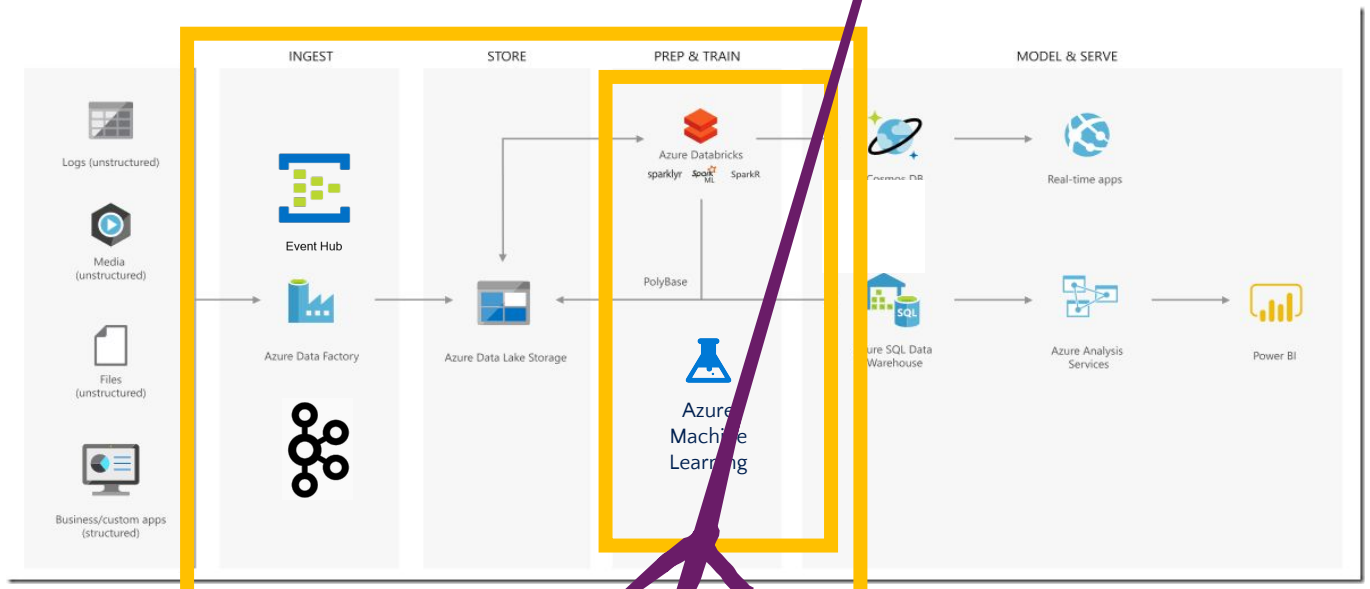


Big Data/ ML Pipelines



@adipolak

Big Data/ ML Pipelines



Check and transform the data

Visualize the model

Balance the data

Explainers

@adipolak

How Microsoft support Responsible AI

ONLINE **FREE** HIGH QUALITY COURSES

INVESTED **1B$** IN OPEN AI

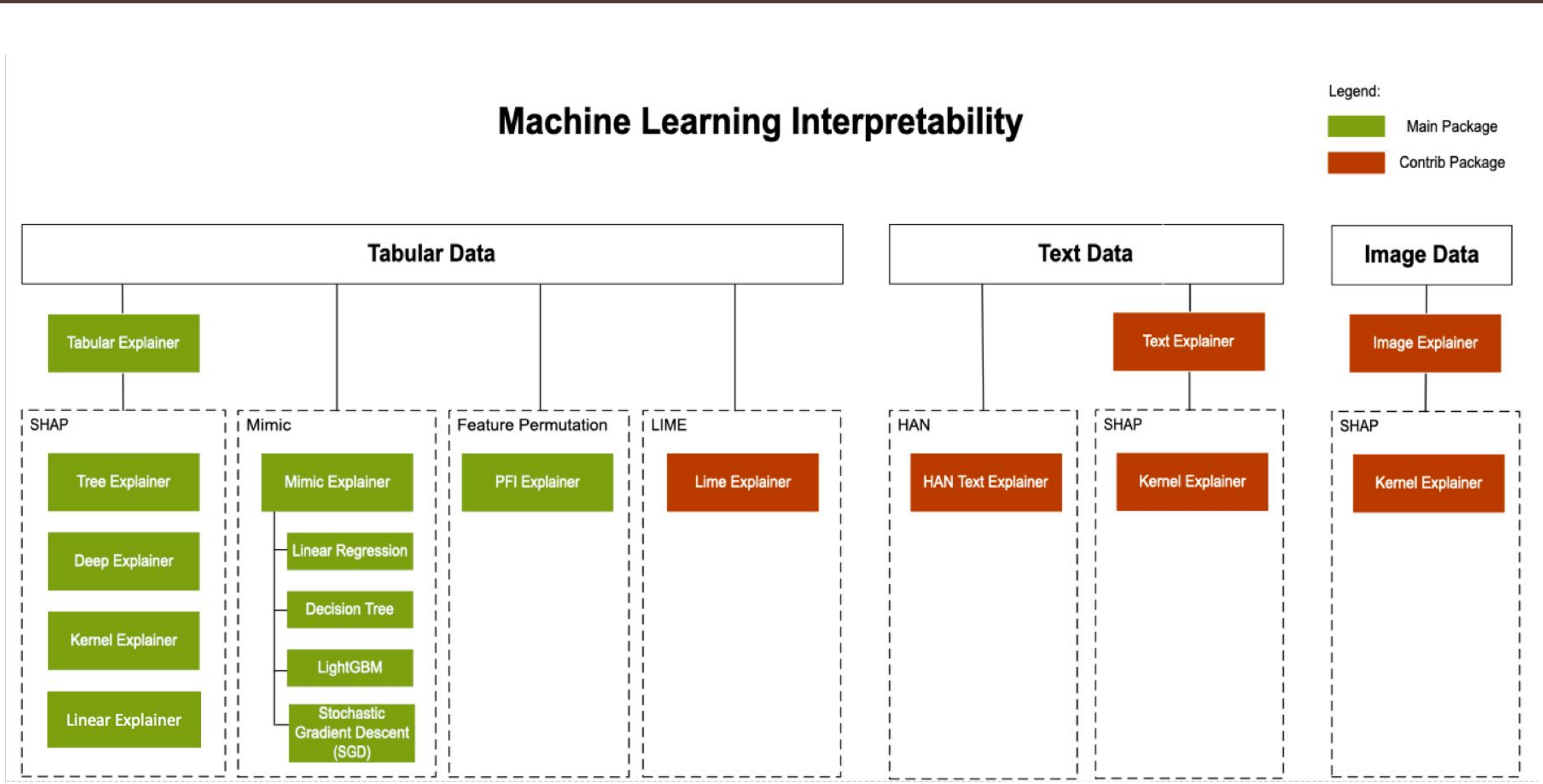
115M$ GRANT FOR AI FOR GOOD

OPEN SOURCE

@adipolak

aka.ms/free-responsible-ai-course

aka.ms/ml-interpretability-tool

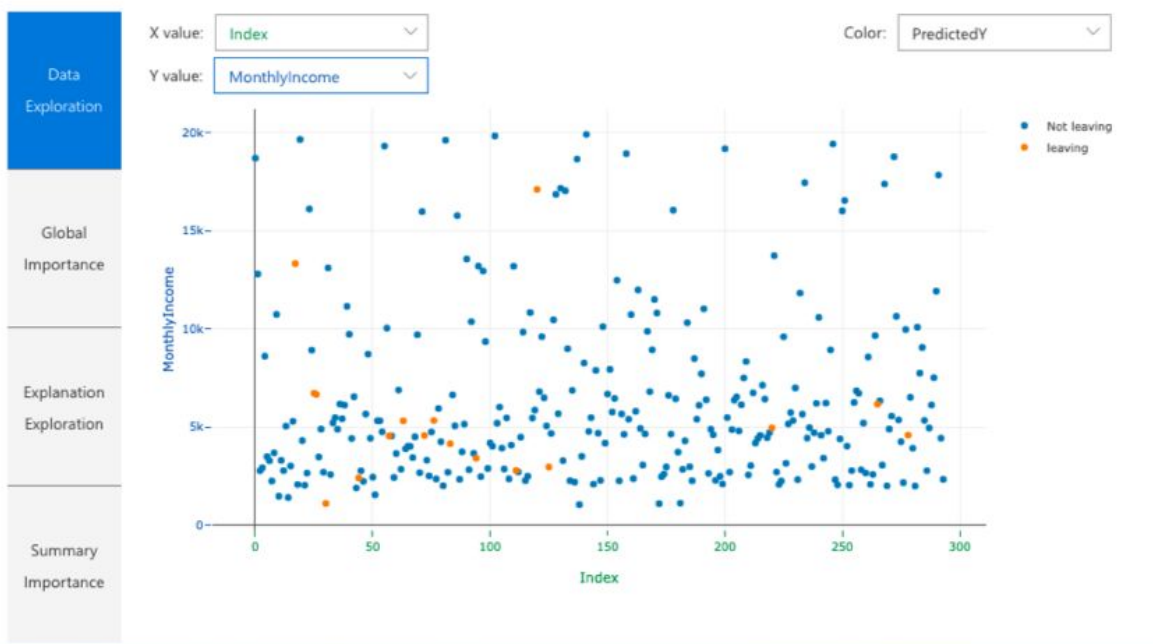


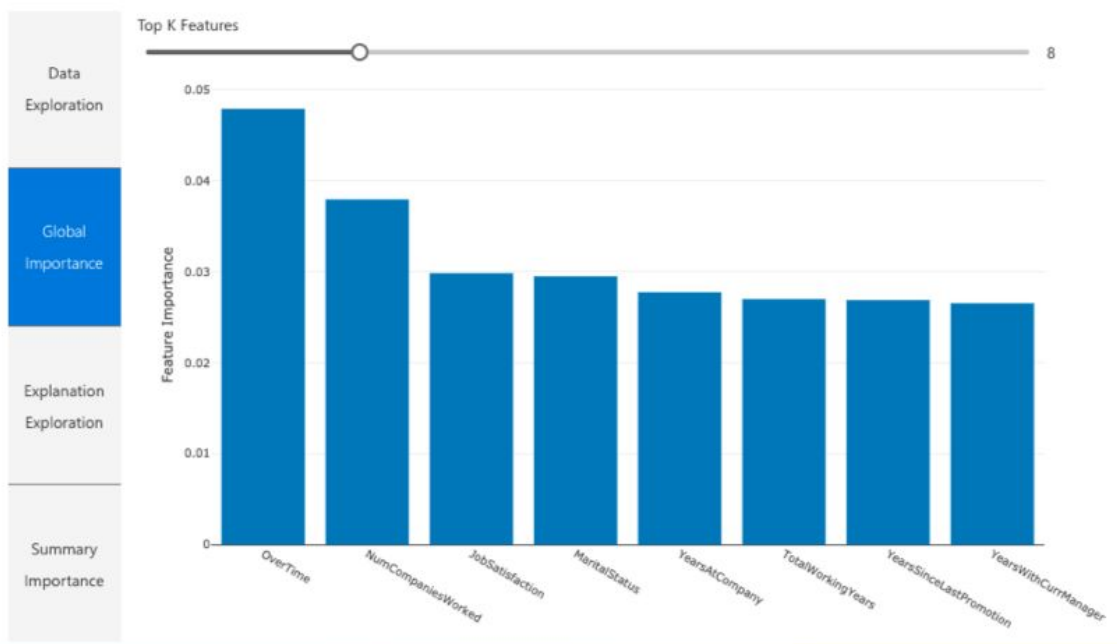
@adipolak

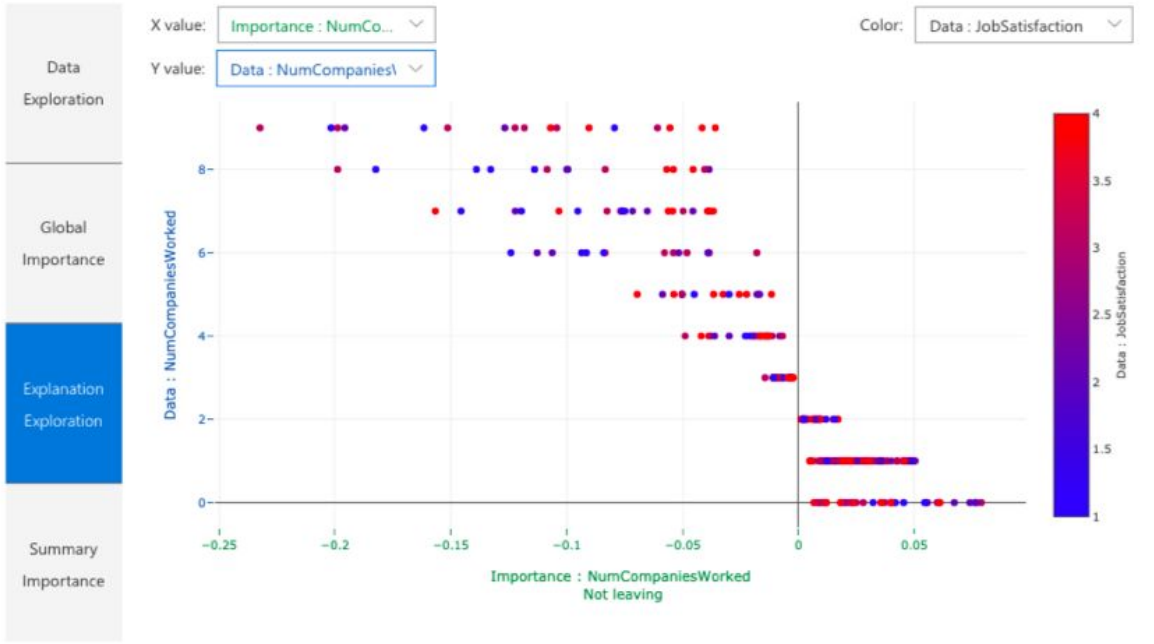
aka.ms/ml-interpretability-tool

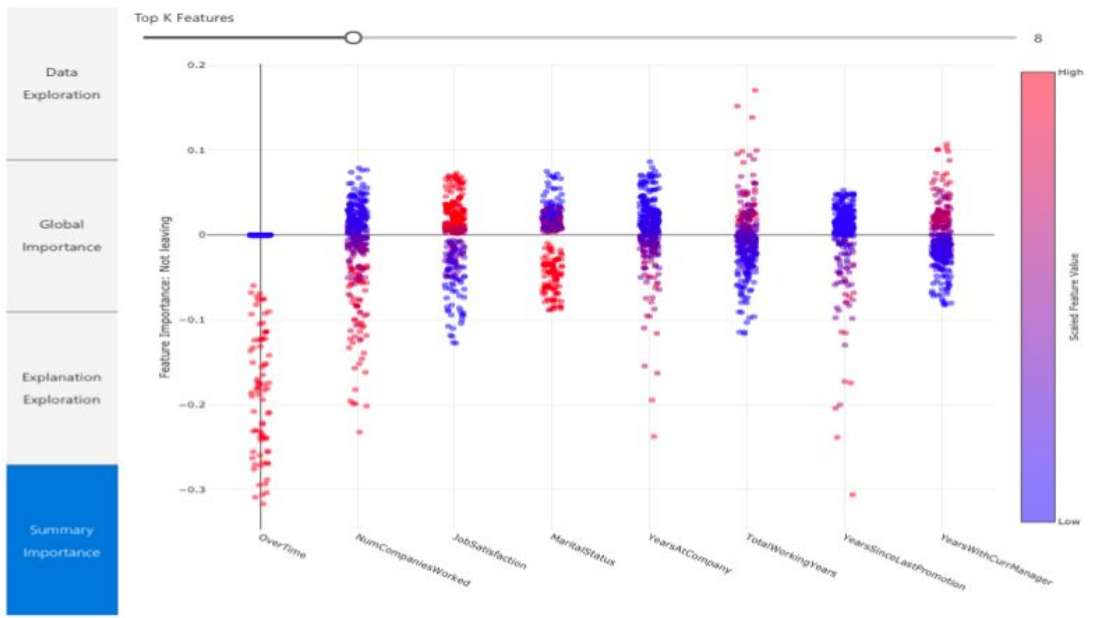
```
# load people dataset
from sklearn.datasets import load_breast_cancer
from sklearn import svm
from sklearn.model_selection import train_test_split
people_data = load_people()
classes = people_data.target_names.tolist()

# split data into train and test
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(people_data.data,
                                                    people_data.target,
                                                    test_size=0.2,
                                                    random_state=0)
clf = svm.SVC(gamma=0.001, C=100., probability=True)
model = clf.fit(x_train, y_train)
```

```
from interpret.ext.blackbox import TabularExplainer

# "features" and "classes" fields are optional
explainer = TabularExplainer(model,
                             x_train,
                             features=people_data.feature_names,
                             classes=classes)
```


Data Exploration
Global Importance
Explanation Exploration
Summary Importance
X value: Index
Y value: MonthlyIncome
Color: PredictedY
Not leaving
leaving
MonthlyIncome
Index


Data Exploration
Global Importance
Explanation Exploration
Summary Importance
Top K Features
8
Feature Importance
OverTime
NumCompaniesWorked
JobSatisfaction
MaritalStatus
YearsAtCompany
TotalWorkingYears
YearsSinceLastPromotion
YearsWithCurrManager


Data Exploration
Global Importance
Explanation Exploration
Summary Importance
X value: Importance : NumCo...
Y value: Data : NumCompanies\
Color: Data : JobSatisfaction
Data : NumCompaniesWorked
Data : JobSatisfaction
Importance : NumCompaniesWorked
Not leaving


Data Exploration
Global Importance
Explanation Exploration
Summary Importance
Top K Features
8
Feature Importance: Not leaving
High
Low
Scaled Feature Value
OverTime
NumCompaniesWorked
JobSatisfaction
MaritalStatus
YearsAtCompany
TotalWorkingYears
YearsSinceLastPromotion
YearsWithCurrManager

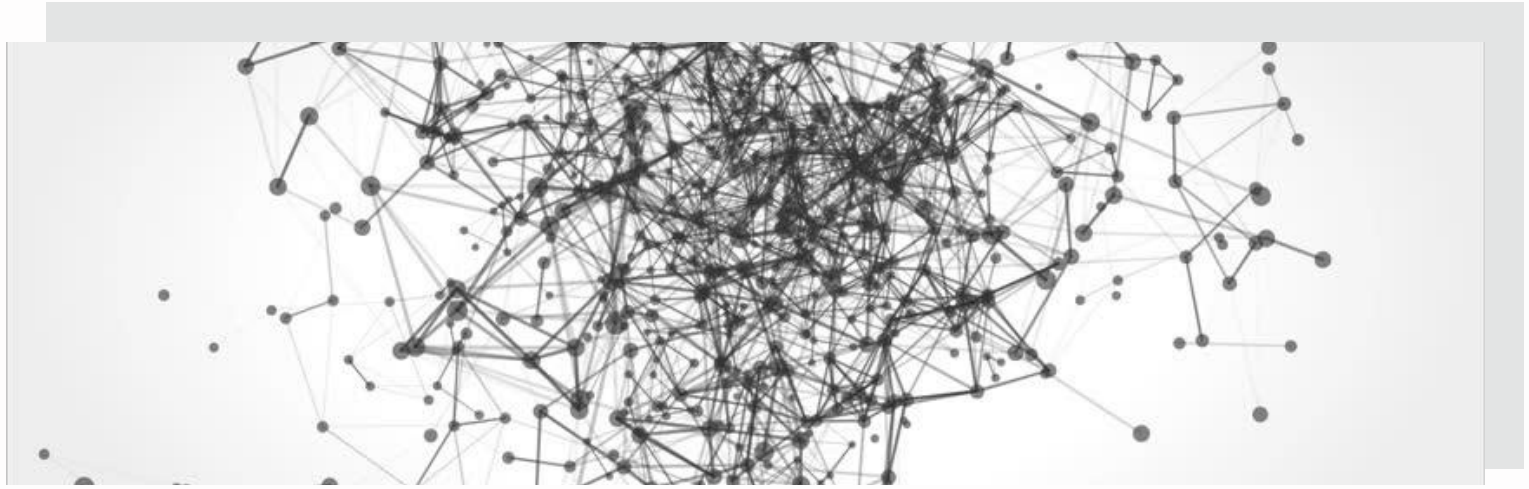
aka.ms/ml-interpretability-tool

Azure Cognitive Services

aka.ms/AA6kexs

@adipolak

Tools



- **Spark Streaming**
- **Spark ML**
- **Spark SQL**
- **MLflow**

@adipolak

):

Demo

Apache Spark ML Pipelines with Cognitive Services

Define MLflow experiment

```
1  val mlflowContext = new MlflowContext()
2  val experimentName = "/Shared/PipelinesExp"
3  val client = mlflowContext.getClient()
4  val experimentOpt = client.getExperimentByName(experimentName);
5  if (!experimentOpt.isPresent()) {
6    client.createExperiment(experimentName)
7  }
8  mlflowContext.setExperimentName(experimentName)
9
10 val run = mlflowContext.startRun("run")
```

Show result

Demo

You are only Good as your Data is

Use explainers

Understand your data

@adipolak

Learn more !

aka.ms/free-responsible-ai-course

aka.ms/twitter_sentiment_analysis

aka.ms/ml-interpretability-tool

aka.ms/ai-for-good-grant

Thank you !

@adipolak

What is Machine learning

- Lifecycle:

- Gather data
- Data preparation – clean it
- Data wrangling
- Data analysis
- Feature extraction
- Train model
- Test model
- Deployment