Deploying Real-Time Decision Sevices using Redis

Tague Griffith, Redis Labs

#MLSAIS12

Why Machine Learning

Teaching a computer, by example, an algorithm that is too complex to program

Machine Learning Problems

Classification

Pick One of a Set

- Spam Detection
- Manufacturing defect detection
- Handwriting analysis

- Decision Trees
- Naïve Bayes
- Logistic Regression

Regression

Score or Rank

- Recommendations
- Likelihood of Purchase

- Linear Regression
- SVM

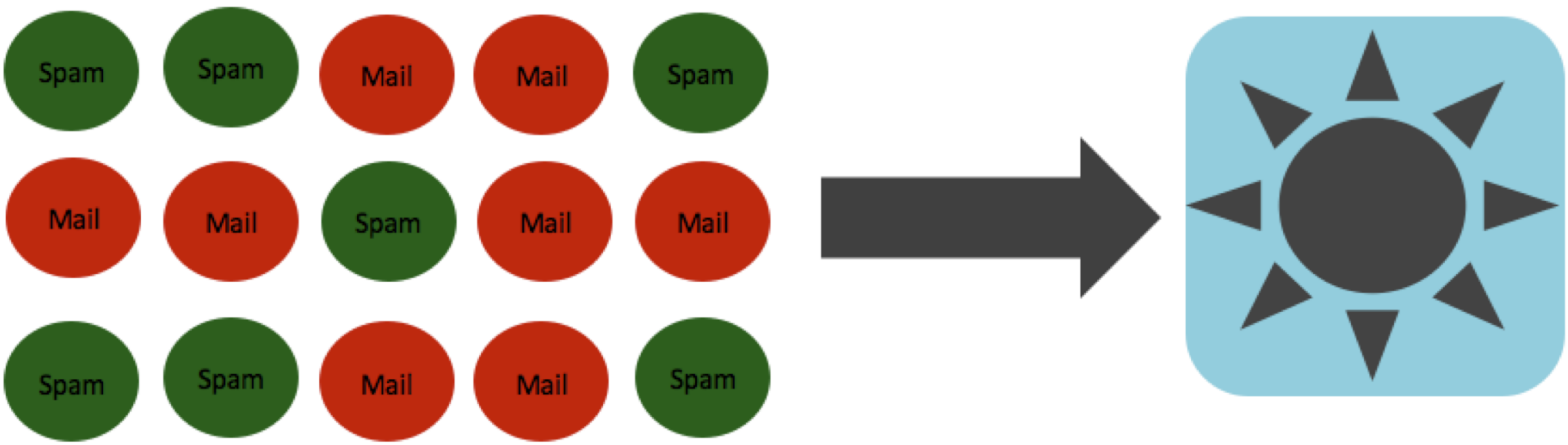
Clustering

Group Similar

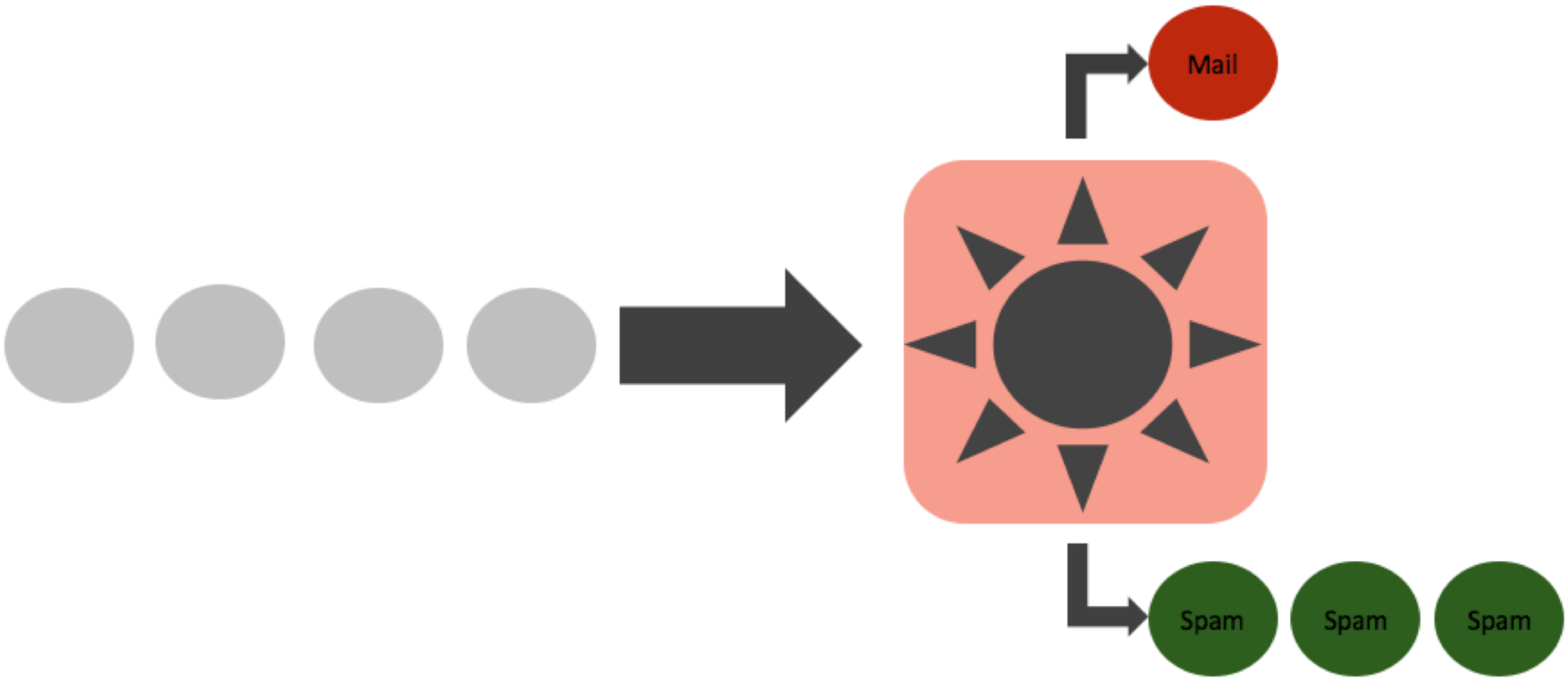
- Find Similar Items
- Customer segmentation
- Cohort detection

- K-Means
- K-Nearest Neighbors
- Hierarchical Clustering

Supervised Learning – Training Spam Classifier



Deploying a Spam Classifier



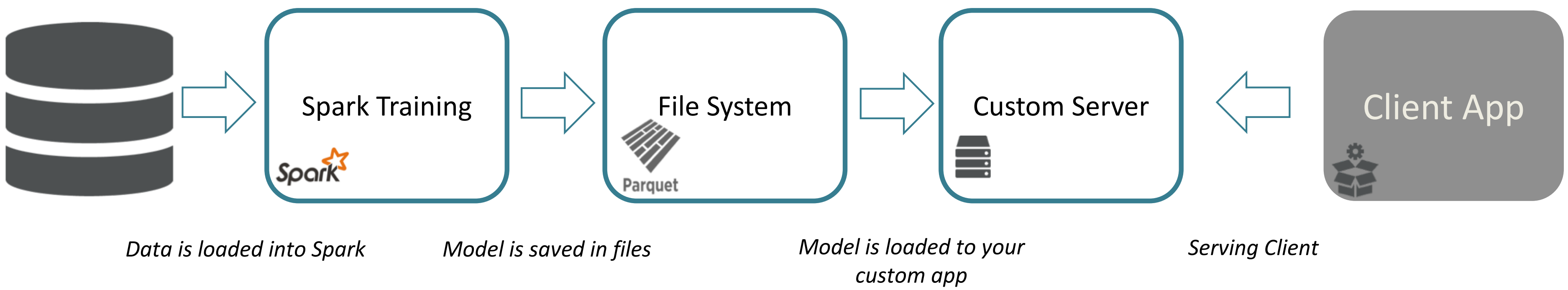
How do we Build these Boxes

¯_(ツ)_/¯

- Building high performance and reliable services are hard, isn't there something we can deploy

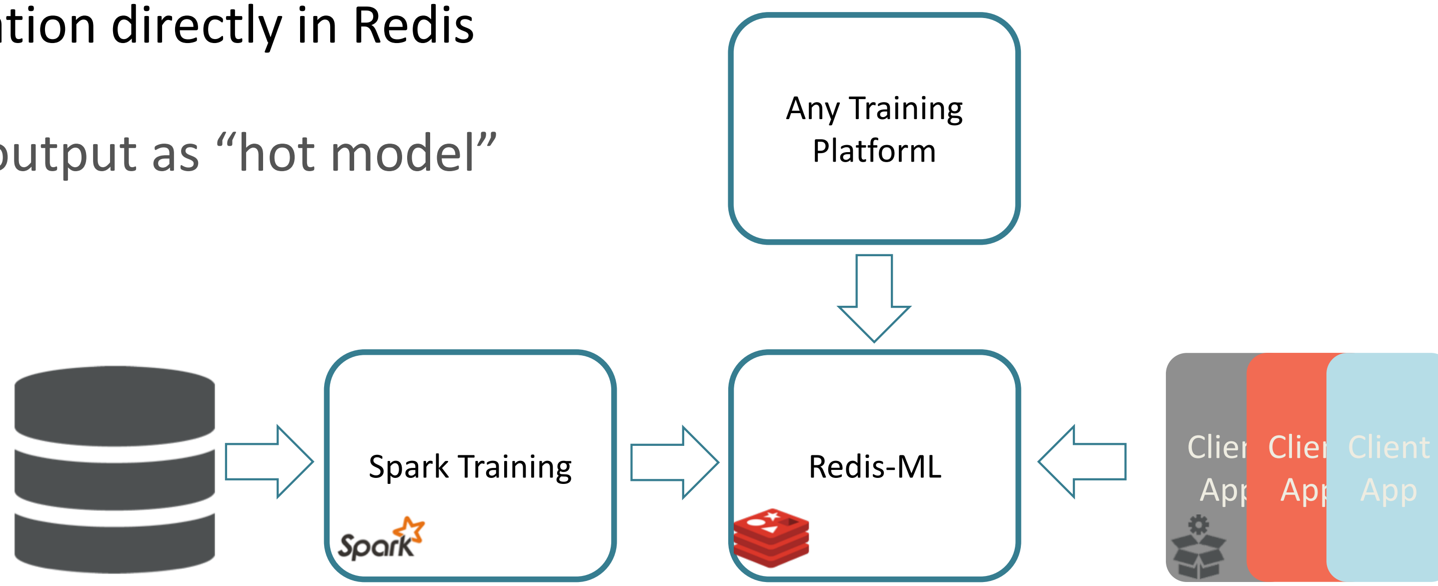
Redis - ML

Typical Spark Application Structure



Redis-ML: Predictive Model Serving Engine

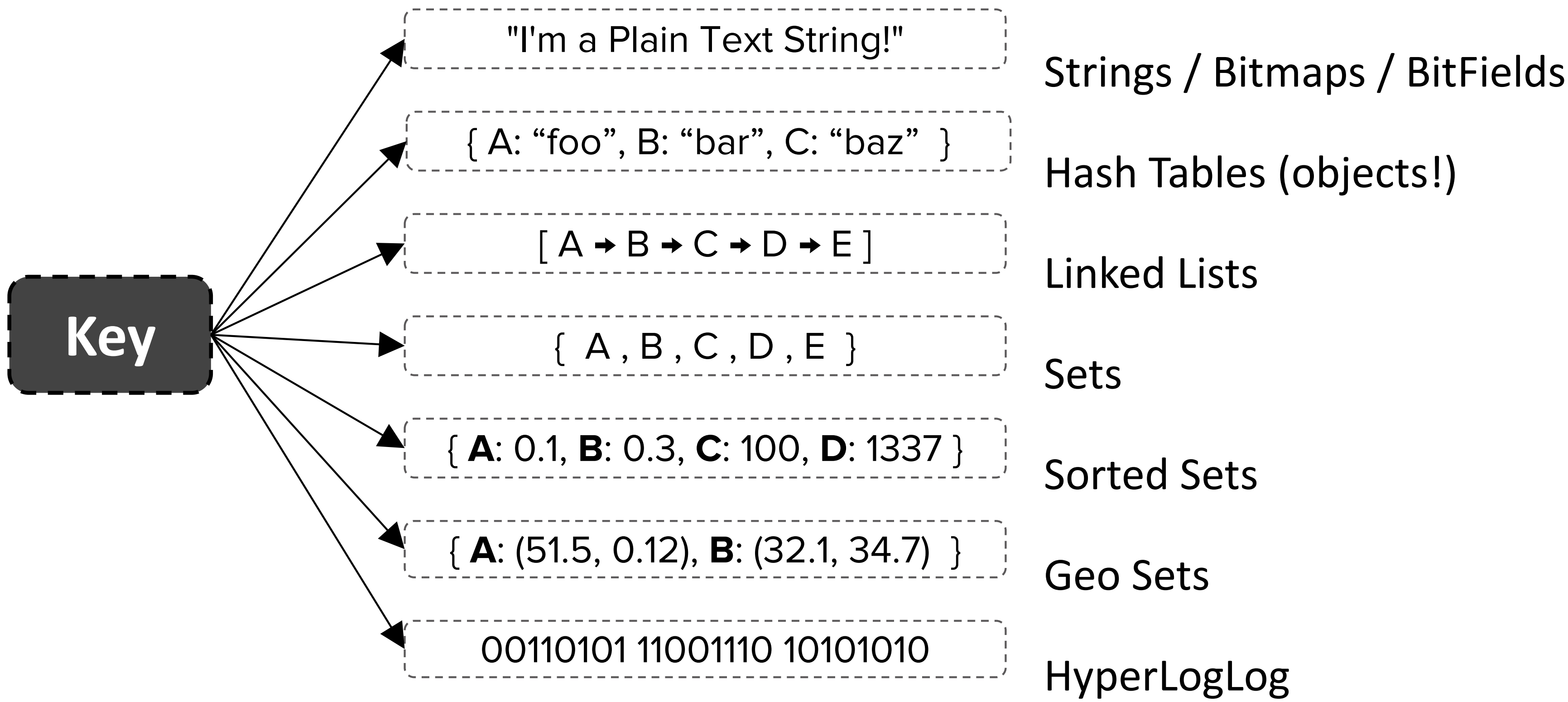
- Predictive models as native Redis types
- Perform evaluation directly in Redis
- Store training output as “hot model”



REmote DIctionary Server

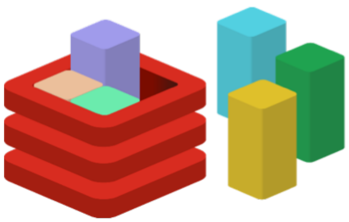
Strings	Hashes	Lists
Sets	Bitmaps	Hyperlog-logs
Sorted Sets	Geo-spatial	Bitfield

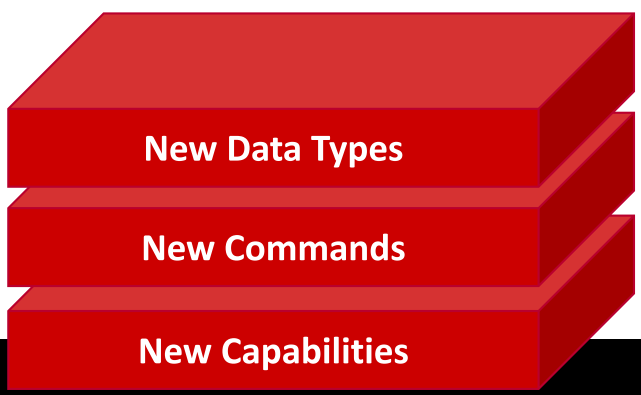
A Quick Recap of Redis



Redis Modules

- Any C/C++ program can now run on Redis
- Use existing or add new data-structures
- Enjoy simplicity, infinite scalability and high availability while keeping the native speed of Redis
- Can be created by anyone





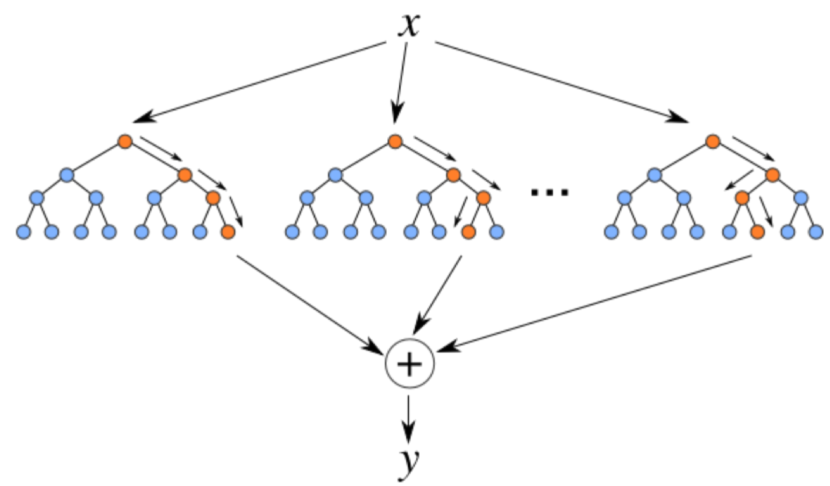
Redis ML Module

Redis Module

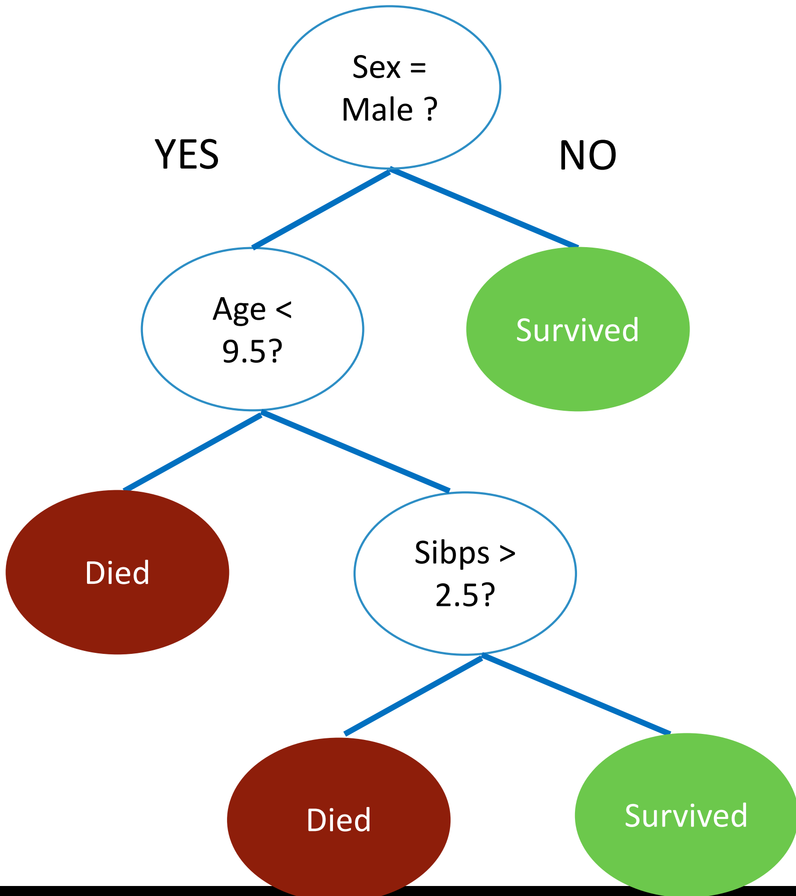
- Tree Ensembles
- Linear Regression
- Logistic Regression
- Matrix + Vector Operations
- More to come...

Random Forest Model

- A collection of decision trees
- Supports classification & regression
- Splitter Node can be:
  - Categorical (e.g. day == “Sunday”)
  - Numerical (e.g. age < 43)
- Decision is taken by the majority of decision trees

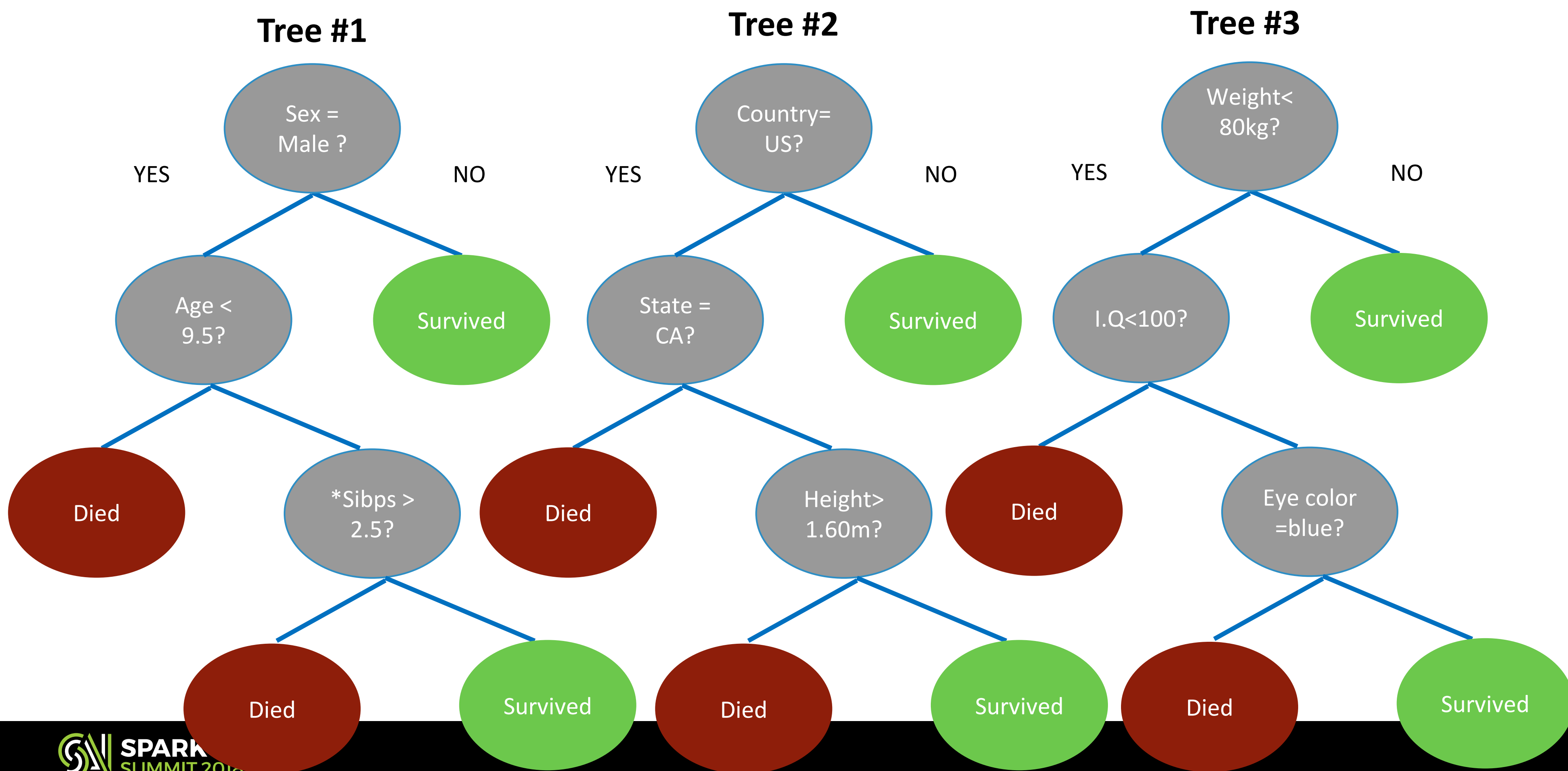


Classic Tree Problem: Titanic Survival



- Passenger Data encoded as feature vect
- ML Algorithm learns the tree rules
  - ID3, CART (RPART), etc.
- Tree rules used to infer results

Titanic Survival: Random Forest



Who Would Survive the Titanic

- John:
  - Male, 34,
  - Married w/ 2 kids (Sibps=3)
  - New York, USA
  - 1.78m, 78kg
  - 110 iq
  - Blue eyes

Mathew:
- Male, 6
- 3 Sisters (Sibps=3)
- New York, USA
- 1.06m, 22.7 kg
- 100 iq
- Brown eyes

Let's use our forest to find out

Redis: Forest Data Type

Add nodes to a tree in a forest:

```
ML.FOREST.ADD <forestId> <treeId> <path>
    [ [NUMERIC|CATEGORIC] <splitterAttr> <splitterVal> ] |
    [LEAF] <predVal>
```

Perform classification/regression of a feature vector:

```
ML.FOREST.RUN <forestId> <features>
    [CLASSIFICATION|REGRESSION]
```

Real World Challenge

- Ad serving company
- Need to serve 20,000 ads/sec @ 50msec data-center latency
- Runs 1k campaigns → 1K random forest
- Each forest has 15K trees
- On average each tree has 7 levels (depth)

Ad Serving costs: Homegrown v. Redis

Cut computing infrastructure

by 97%

Homegrown

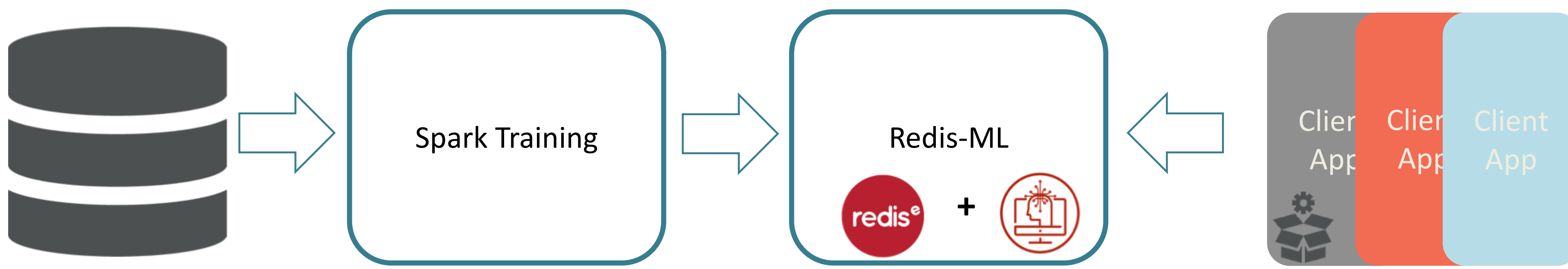
1,247 x c4.8xlarge

35 x c4.8xlarge

Summary

- Train with Spark, Serve with Redis
- 97% resource cost serving
- Simplify ML lifecycle

- Redis[e] (Cloud or Pack):
  - Scaling, HA, Performance
  - PAYG – cost optimized
  - Ease of use
  - Supported by the teams who created Spark and Redis



Thank you!