Spark: Data Science as a Service

Sridhar Alla, Shekhar Agrawal
Comcast

SPARK SUMMIT EAST 2017

Who we are

- Sridhar Alla
  Director, Data Engineering, Comcast
  Architecting and building solutions on a big data scale
  sridhar_alla@cable.comcast.com

- Shekhar Agrawal
  Director, Data Science, Comcast
  Data science on a big data scale.
  shekhar_agrawal@cable.comcast.com

Agenda

- Sample Data Science Use cases
- Real world Challenges
- Introduction to Sparkle – Our Solution to the real world challenges
- Integration of Spark with Sparkle
- How we use Sparkle in Comcast
- Q & A

Data Science Use Case

- Churn Models
- Price Elasticity
- Geo Spatial Route Optimization
- Direct Mail Campaign
- Customer call Analytics
- many more ………….

Real World Challenges

- We store and process massive amounts of data, still lack critical ability to stitch together pieces of data to make meaningful predictions. This is due to
  - Massive data size
  - Lack of service level architecture
- Multiple teams working on the same dataset
  - This increases development time because everyone has to process/feature engineer same dataset

Our Data

- 40PB in HDFS capacity and 100s of TBs in Teradata space
- ~1200 data nodes in total in Hadoop and Spark clusters
- 100s of models
  - Logistic regression
  - Neural Networks
  - LDA and other text analytics
  - Bayesian Networks
  - Kmeans
  - Geospatial

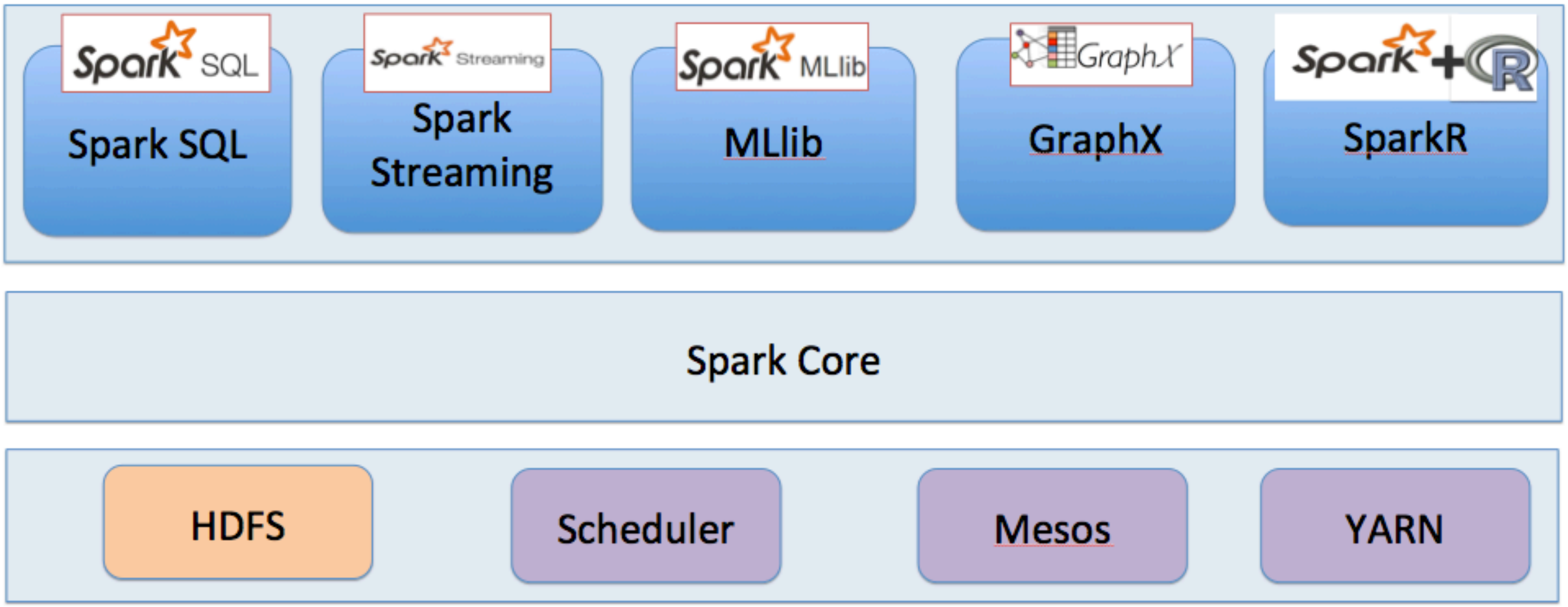
What we need is

- A Central Processing System
  - Highly Scalable
  - Persisted and Cached
  - SQL capabilities
  - Machine Learning capabilities
  - Multi Tenancy
  - Access through low level language

What we built

- Perpetual Spark Engine
- RESTful API to control all aspects
- Connectors to
  - Cassandra, Hbase, MongoDB etc
  - Teradata, MySQL etc
  - Hive
  - ORC, Parquet, text files
- Role based control on who sees what
- Integration with modeling using Python, R, SAS, SparkML, H2O

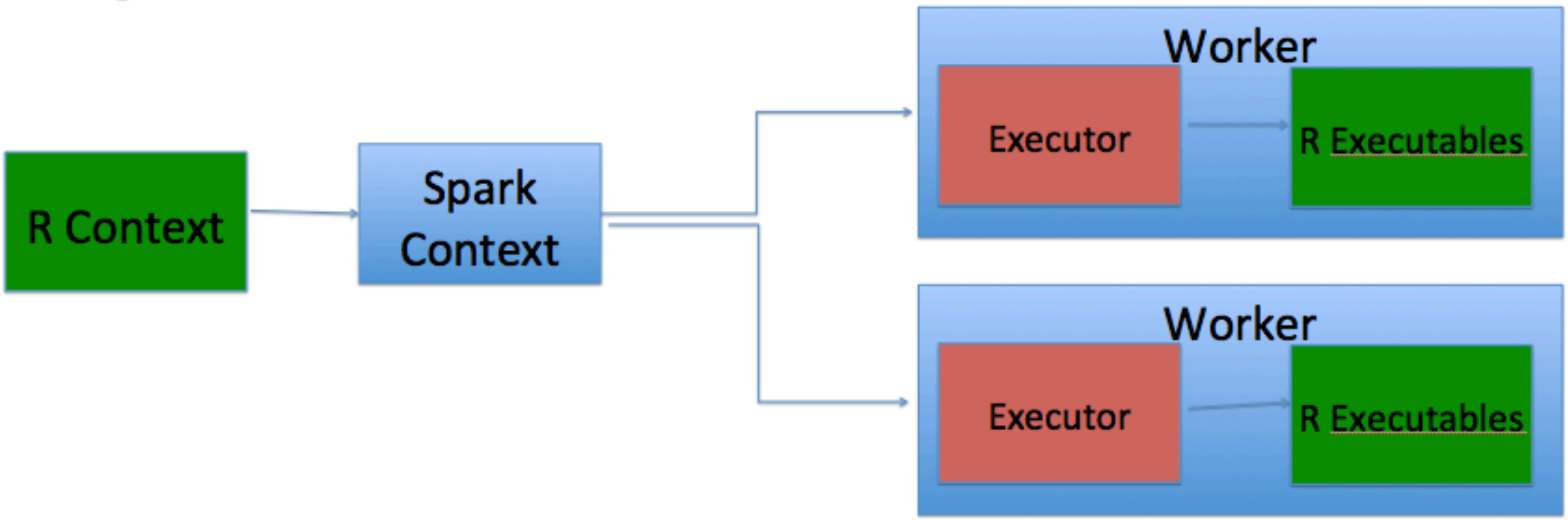
Spark Stack



Spark SQL
Spark Streaming
MLlib
GraphX
SparkR
Spark Core
HDFS
Scheduler
Mesos
YARN

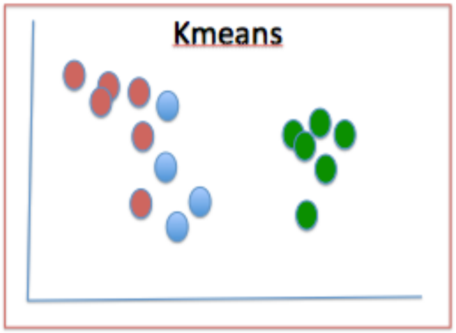
SparkR

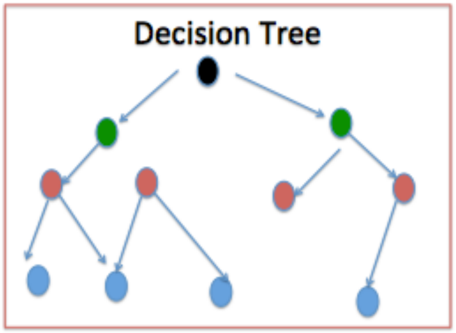
- Enables using R packages to process data
- Can run Machine Learning and Statistical Analysis



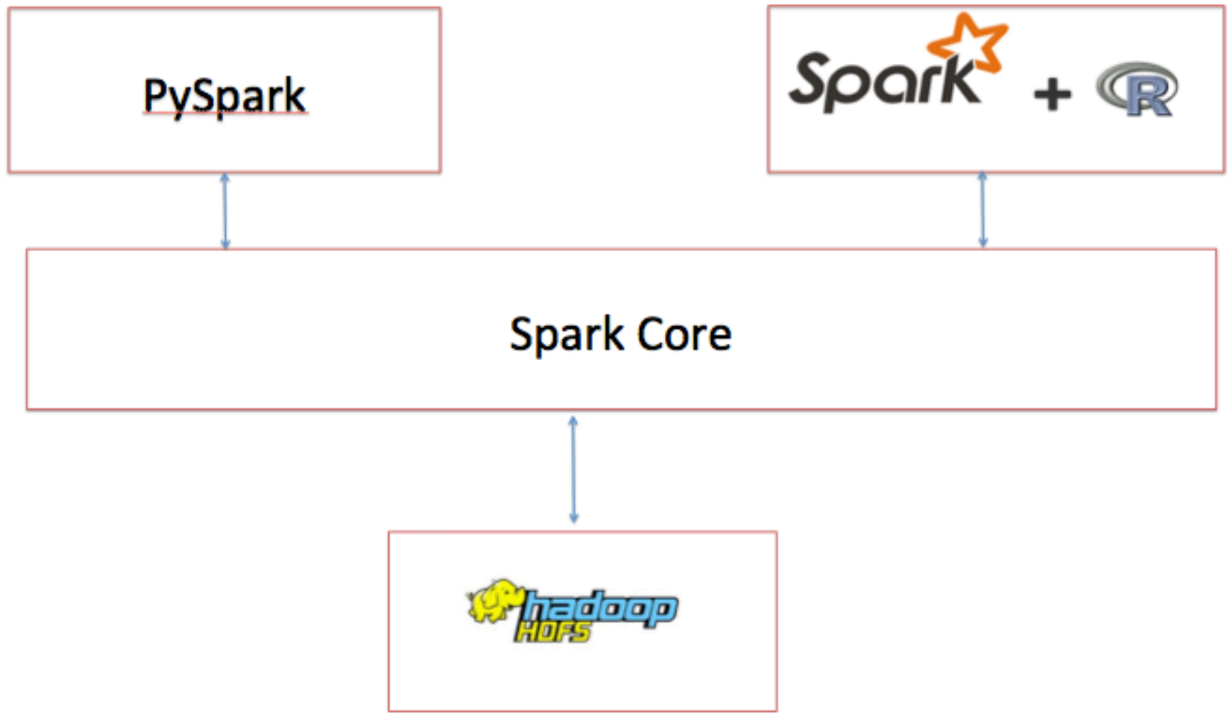
Spark MLlib

- Implements various Machine Learning Algorithms
- Classification, Regression, Collaborative Filtering, Clustering, Decomposition
- Works with Streaming, Spark SQL, GraphX or with SparkR.

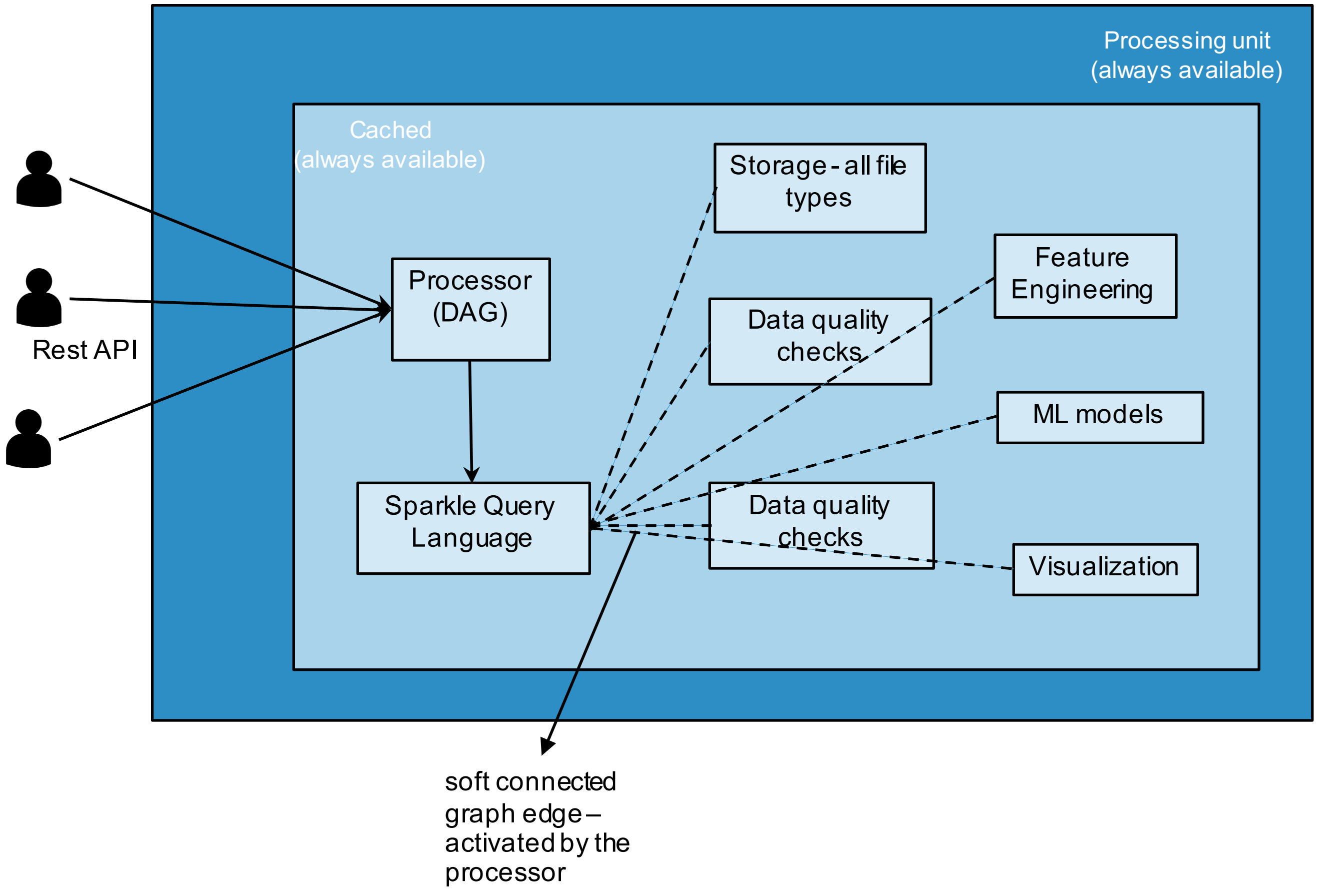




Using PySpark & SparkR

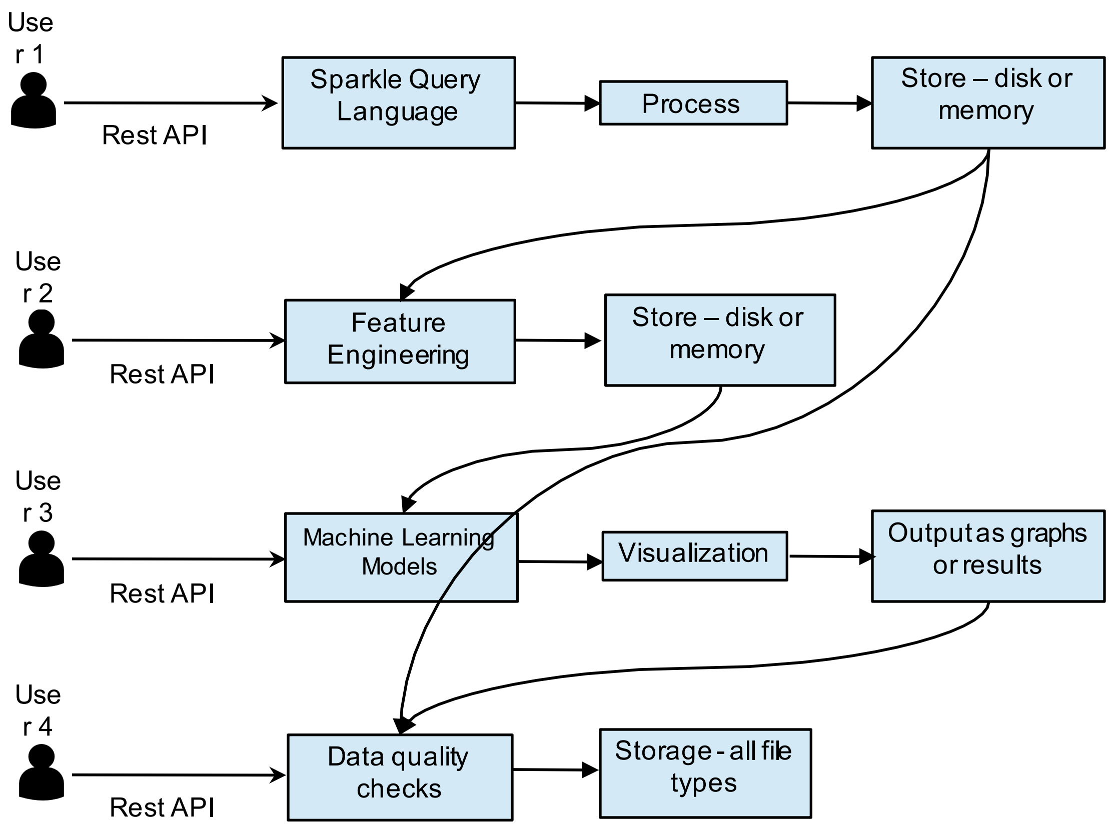


Introduction to Sparkle



SPARK SUMMIT EAST 2017

What can be done using Sparkle



SPARK SUMMIT EAST 2017

Sample Rest API

Processing → Rest API

Feature Engineering → Rest API

Modeling → Rest API


```
{
  "jobType": "nautilusPathsJob",
  "jobId" : "JobId4",
  "rosettaTableName":"base.adm_meld_201607",
    "startTime":"2016-01-01 00:00:00",
      "endTime":"2016-02-01 00:00:00",
   "eventId":"ANY",
   "appendToEventId":"",
   "minAccounts": 1,
     "accountFilters": "ALL",
     "eventRules": {
            "condition": "OR",
            "rules" : [
              {
              "ruleType" : 2,
              "firstEventId": "ER.*",
              "secondEventId": "IVR.*",
              "op" : "gt",
              "threshold": 3,
              "timeGap" : 166400,
              "generateRuleSequences":true,
              "overlappingSequences":true,
              "exactMatchingEvents":true
              }
            ]
          }
}
```


SPARK SUMMIT EAST 2017

Sample Rest API

Processing → Rest API

Feature Engineering → Rest API

Modeling → Rest API


```
{
  "jobType": "motoJob",
  "jobId" : "moto1",
  "campaignName" : "NE_Explore",
  "campaignCondition":
            {
    "division":"NORTHEAST DIVISION",
    "channels":"tbd_mailed"
   },
  "jobStage" : "updateNode",
  "category":{"romi":">=0","connects":"2010"}
}
```

Who can use Sparkle

Statistician

Dev Ops

Validation

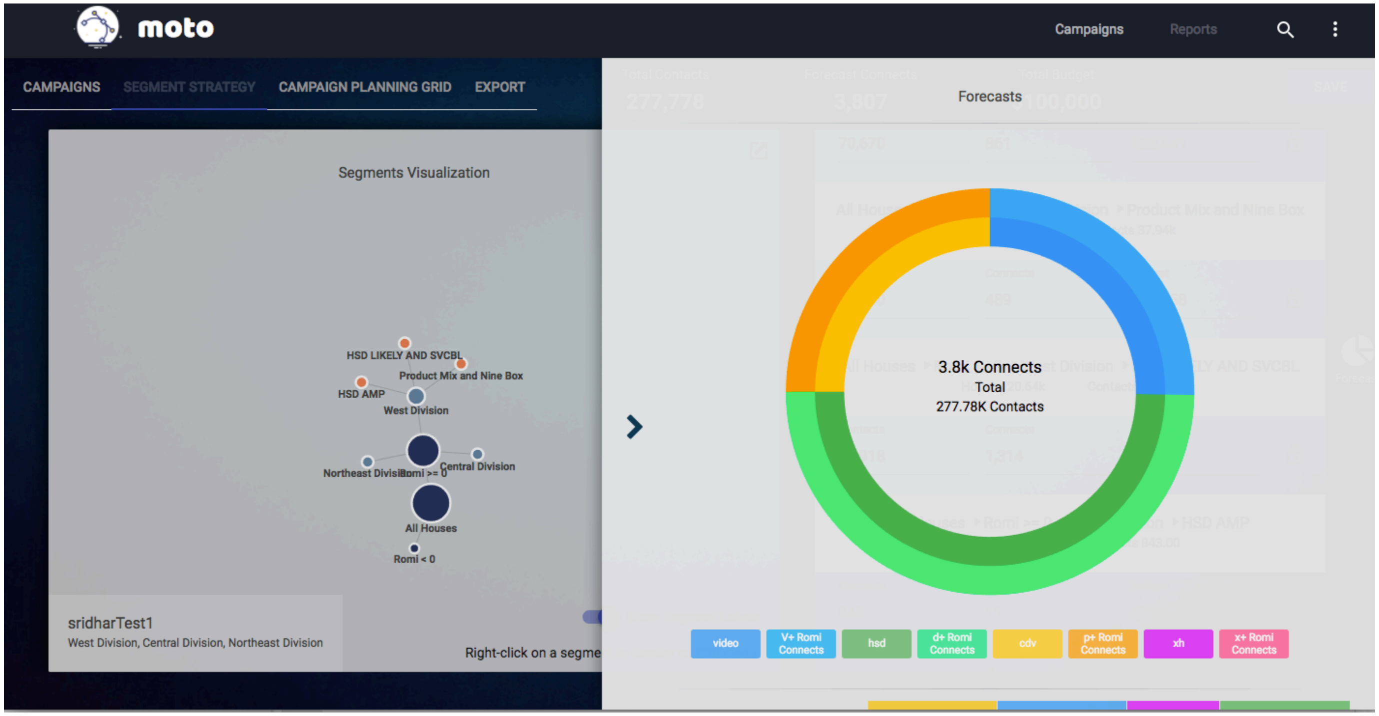
Modeler

Data
Engineer

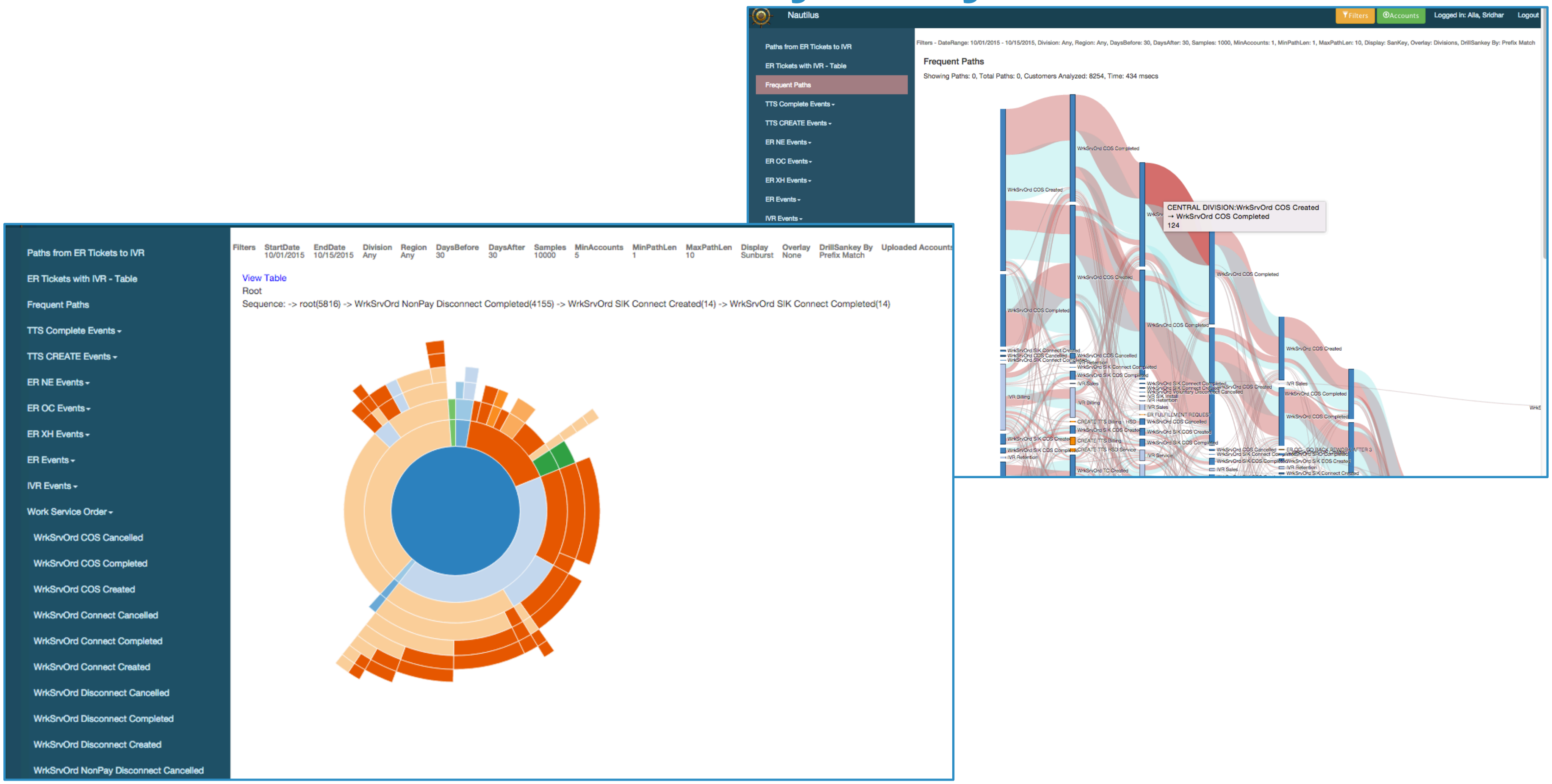
Data
Scientist

Anyone who know how to use Rest API can use Sparkle. This also decreases development time by high degree

• **MOTO – Direct Mail Campaign Optimization**



- **Nautilus – Customer Journey Analytics**



We are hiring!

- Big Data Engineers (Hadoop, Spark, Kafka…)
- Data Analysts (R, SAS…..)
- Big Data Analysts (Hive, Pig ….)

jobs.comcast.com

Thank You.

Contact information or call to action goes here.

Sridhar Alla
Director, EBI Data Science
Sridhar_Alla@cable.comcast.com
617.512.9530

Shekhar Agrawal
Director, EBI Data Science
Shekhar_Agrawal@cable.comcast.com
571.267.9239

SPARK SUMMIT EAST 2017

Data Science Initiatives

- Customer Churn Prediction
- Click-thru Analytics
- Personalization
- Customer Journey
- Modeling
- Anomaly Detection

Anomaly Detection

- Identification of observations which do not conform to an expected pattern.
- Ex: Network Intrusion Detection, Spikes in operational data, Unusual usage activity.

Popular Algorithms

- Unsupervised
  - KMeans
  - DBScan
- Supervised
  - HMM
  - Neural networks

KMeans Clustering

- Clustering is an unsupervised learning problem
- Groups subsets of entities with one another based on some notion of similarity.
- Easy to check if a new entity is falling outside known groups/clusters

Sample Code

```
import org.apache.spark.mllib.clustering.KMeans
import org.apache.spark.mllib.linalg.Vectors

val lines = sc.textFile("training.csv")
val data = lines.map(line => line.split(",").map(_.trim))
val inData = data.map{(x) => (x(3)) }.map(_.toLong)
val inVector = inData.map{a => Vectors.dense(a)}.cache()
val numClusters = 3
val numIterations = 100
val kMeans = new KMeans().setMaxIterations(numIterations).setK(numClusters)
val kMeansModel = kMeans.run(inVector)

// Print cluster index for a given observation point
var ci = kMeansModel.predict(Vectors.dense(10000.0))
var ci = kMeansModel.predict(Vectors.dense(900008830.0))
```

Sample Code (R):

```
library('RHmm')
indata <- read.csv(file.choose(), header = FALSE, sep = ",", quote = "\"", dec = ".")
testdata <- read.csv(file.choose(), header = FALSE, sep = ",", quote = "\"", dec = ".")
dataSets <- c(as.numeric(indata$V4))
dataSetModel <- HMMFit(dataSets, nStates=3)
testdataSets <- c(as.numeric(testdata$V4))
tVitPath <- viterbi(dataSetModel, testdataSets)

#Forward-backward procedure, compute probabilities
tfb <- forwardBackward(dataSetModel, testdataSets)

# Plot implied states
layout(1:3) dataSet
plot(testdataSets[1:100],ylab="StateA",type="l", main="dataSet A")
plot(tVitPath$states[1:100],ylab="StateB",type="l", main="dataSet B")
```

Add Slides as Necessary

- Supporting points go here.