Practical machine Learning with MLlib

By
N.Ravitha Rajalakshmi
Assistant Professor
PSG College of Technology

Machine Learning

Constructing models that learn from and make predictions on data

- Deriving Knowledge from data
- Supports wide range of applications
- Data ?
  - Complexity (Structure)
  - Size

Trends

- Rapid growth of Massive datasets
  - Genomics
  - Online user activity
  - Data from sensors
- Pervasiveness of distributed and cloud computing Infrastructure
  - Provide Storage and computational resources for processing

Applications

- Recommendation systems
- Spam Filtering
- Speech Recognition
- Face Recognition
- Link Prediction
- Protein Structure prediction (given acid sequence -> 3d protein structure)

Terminologies – Machine Learning

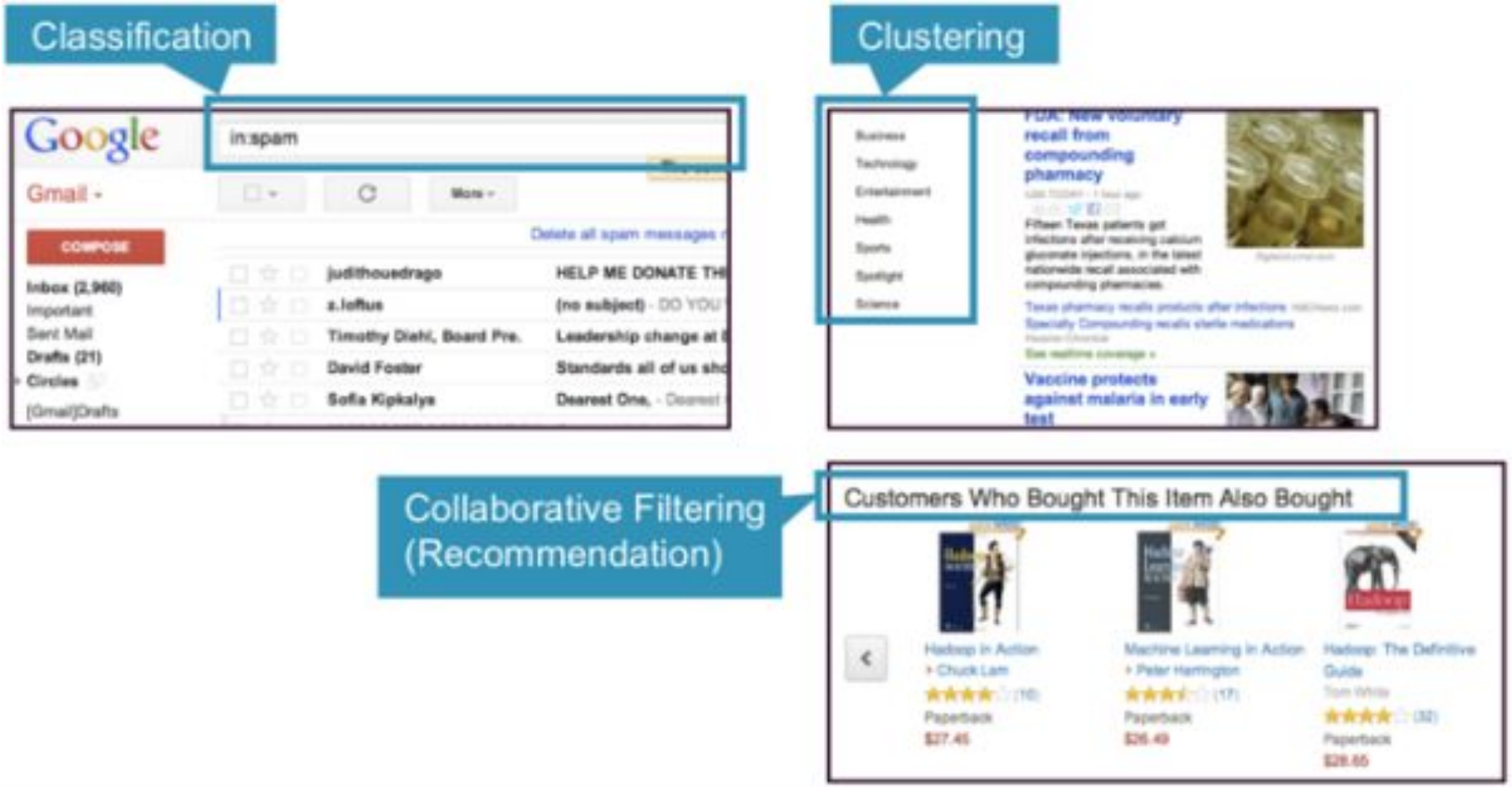
- Learn from observation
- Observations – items / entities used for learning
- Features /estimators– attributes used to represent the observation
- Labels – categories assigned to the observations.

- Training and Test Data : Observations used to train and evaluate the learning algorithm
  - Training Data: Provided to algorithm for training
  - Validation Data : Test data is used to evaluate the model

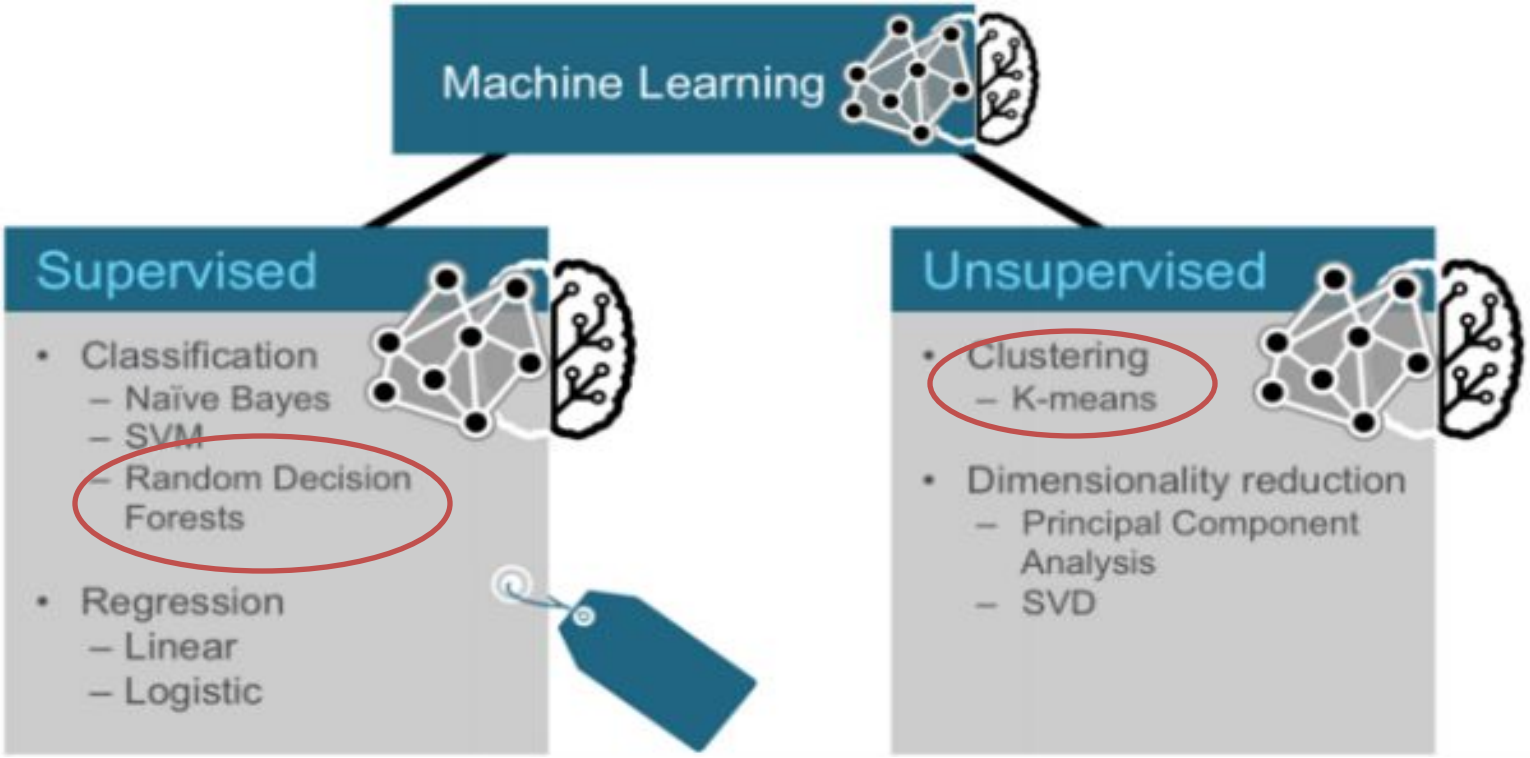
Common Learning Settings

- Supervised : Learns from labeled observation
  - Find the mapping from observations to labels.
- Unsupervised :Learn from unlabeled observation
  - Find latent structure from features alone and groups the observations
  - Find Hidden Patterns
  - Could be employed as preprocessing stage for supervised classifier

Machine Learning – Virtually Applied Everywhere



Machine Learning Algorithms



Challenge

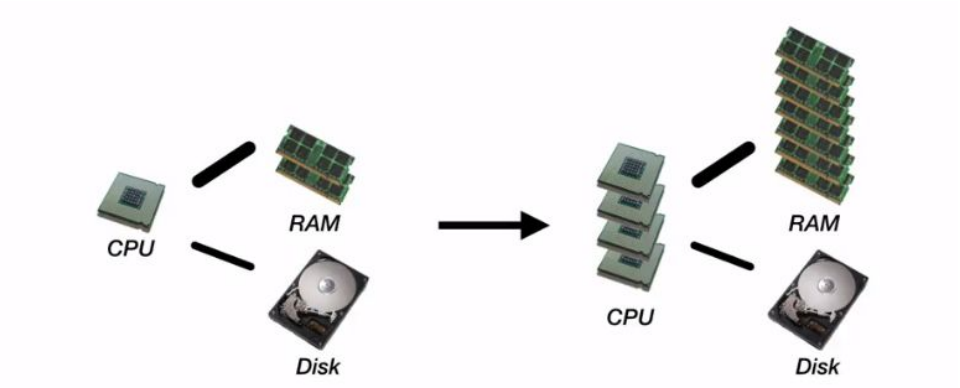
- Scalability for common machine learning tasks
- To deal with massive datasets
  - Distributed Machine Learning algorithms
  - Data preprocessing technique

Why Distributed Computing ??

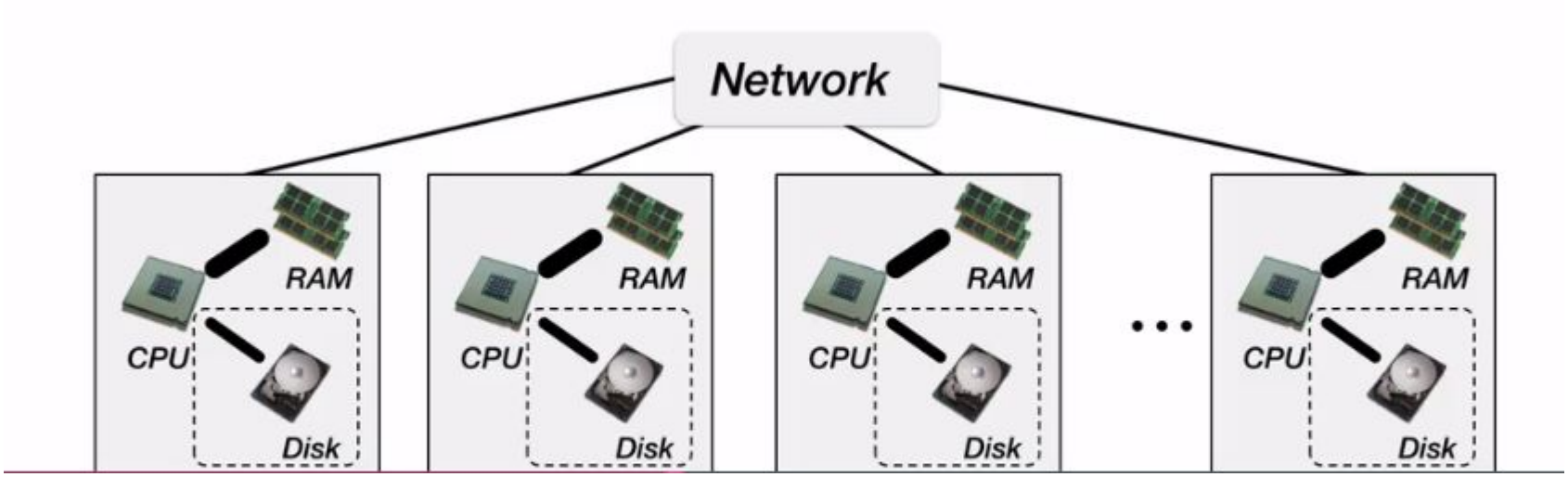
- Why can't Traditional Tools (Matlab, R, Excel) cannot be used for processing large datasets
  - They typically run on single machine.
  - Need more hardware to store / process data

What Options ??

- Scale – up the machine (large machine)
  - Good Idea ! Actually it works faster
  - But need Specialized hardware (expensive)
  - Scaling can be done to a certain extent



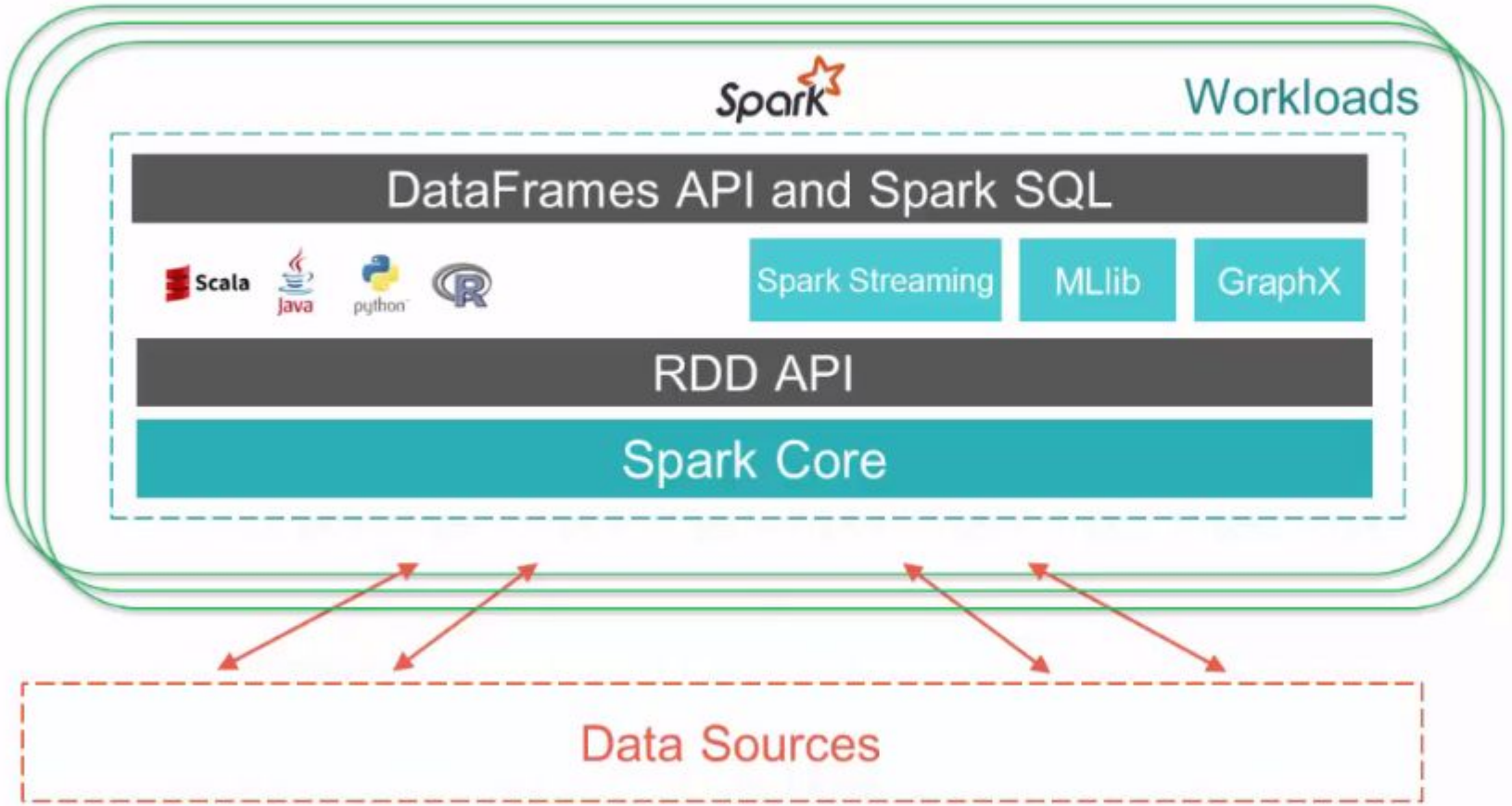
- Scale out
  - Many small machine connected them over network in distributed setting
    - Better alternative , as nodes can easily be added
    - Commodity hardware
    - Network Communication , Software Complexity



Apache Spark

- Open source cluster computing engine
- Why spark for large scale machine learning?
  - Fast iterative computation
  - Communication primitive
  - Provides API for scala, python and java
  - Interactive shell
  - Many high level libraries are available for building machine learning pipelines

Components of Spark



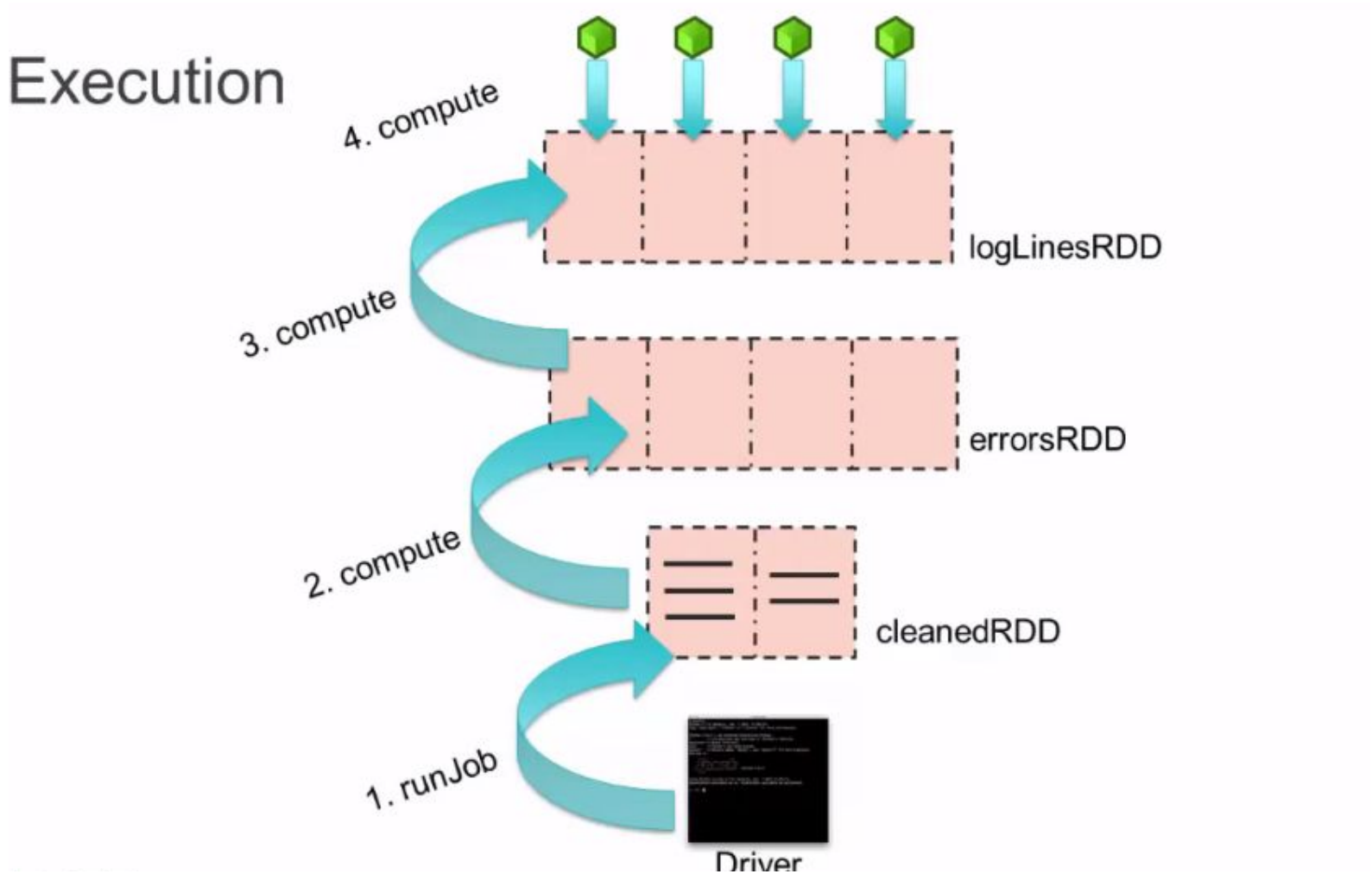
- Spark core , RDD API – low level access to spark functionality
- MLlib, Streaming, GraphX , DataFrames API and SparkSQL – high level operation (top of RDD API)

Resilient distributed Datasets

- Collection of objects distributed across the cluster. (Divide them into partitions and distribute it across nodes)
- In order to manipulate the data, two operations are typically supported: action and transformation.

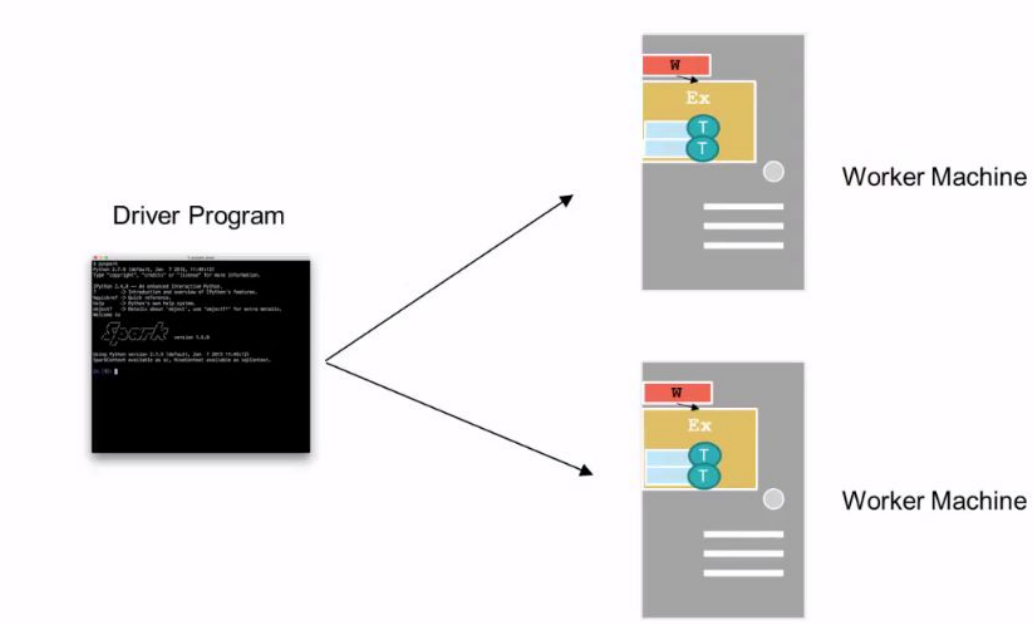
- Apache Spark uses lazy evaluation
- Action causes execution to begin. Launch spark jobs and related transformations are computed.
- Otherwise the operations are represented in a Directed acyclic graph

Data Lineage



Typical Workflow in Spark

- Driver creates DAG (Directed Acyclic Graph) for work to be done and sends it to the worker nodes in the cluster . Cluster will return results to driver program



Create RDD in Python

- Calling parallelize method on spark context

  wordRDD=sc.parallelize([“cats”,”dogs”,”fish”]
)

- Create RDD from local text file

  wordRDD=sc.textfile(“/path/to/ReadMe.md”)

MLlib

- Machine learning package available in Spark.
- It is shipped with Spark 0.8.
- Started as a project in UC Berkeley AMPLab.
- It consists of common learning algorithms and utilities including classification, regression, clustering, collaborative filtering, dimensionality reduction.

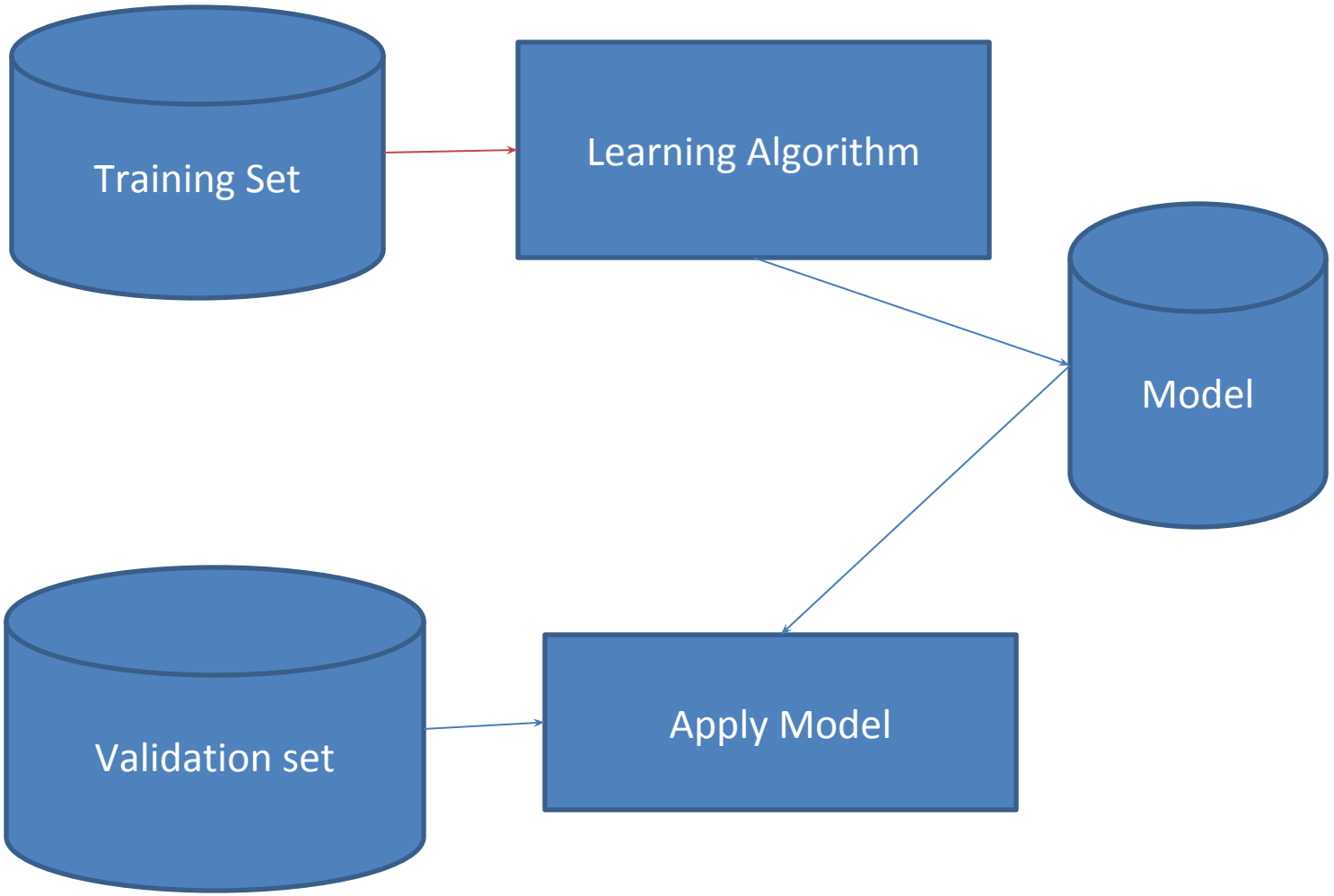
Why MLlib?

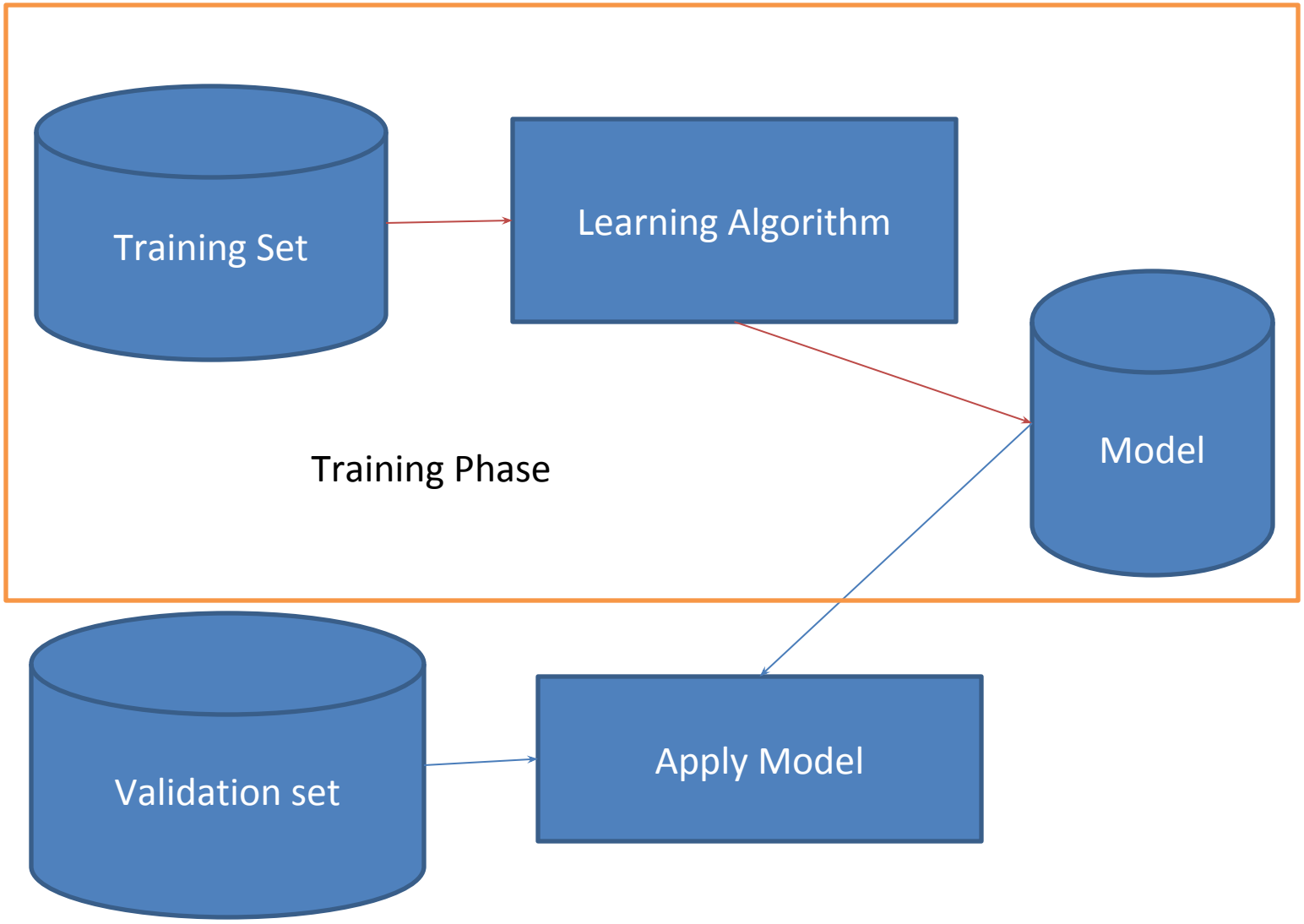
- Scalability
- Better Performance
- Usability

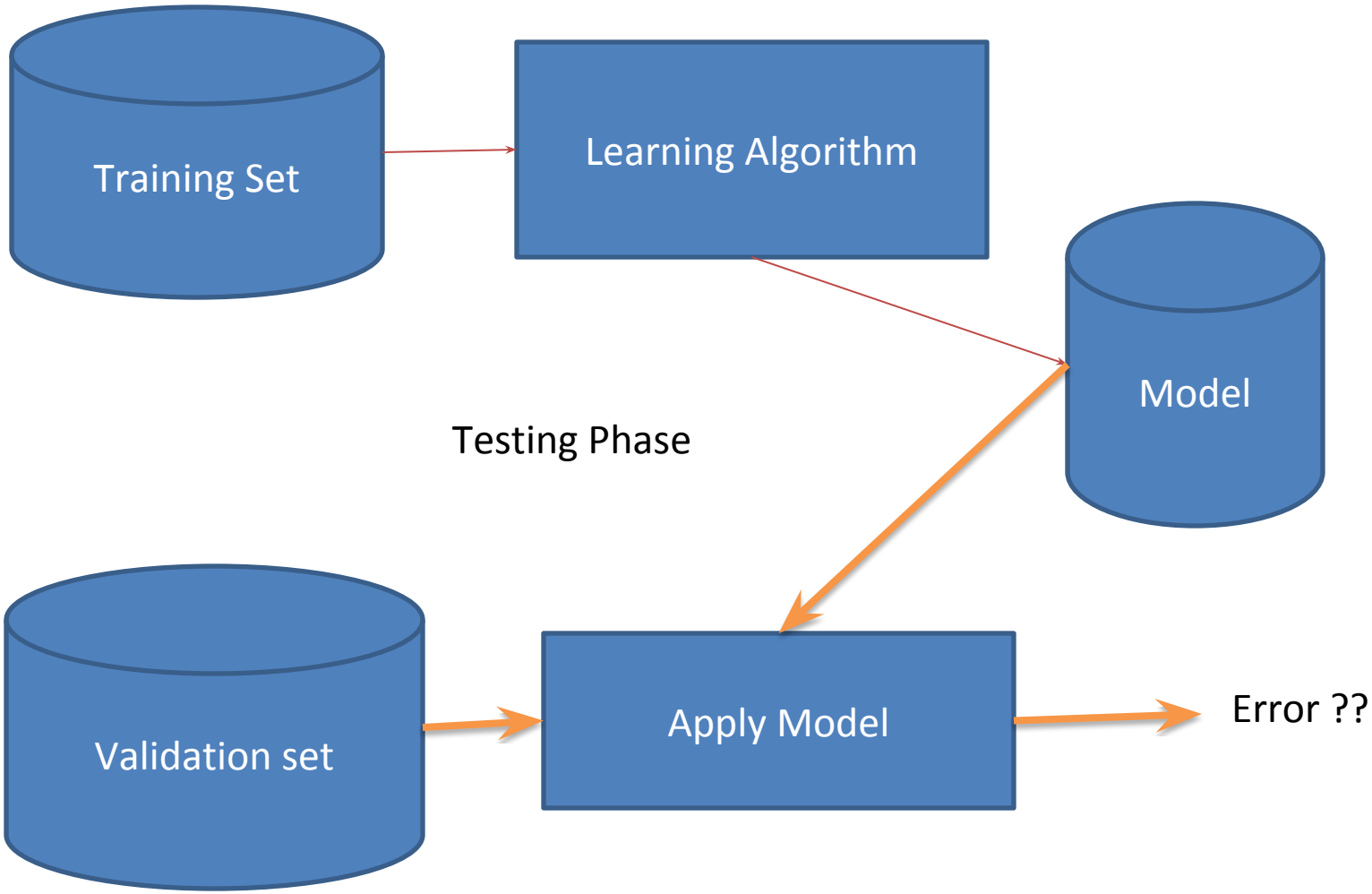
MLlib

- Machine learning has to be easy and scalable
  - Capable of learning from large datasets.
  - Easy way to build machine learning applications.

Classifier - Phases




Training Set
Learning Algorithm
Model
Training Phase
Validation set
Apply Model

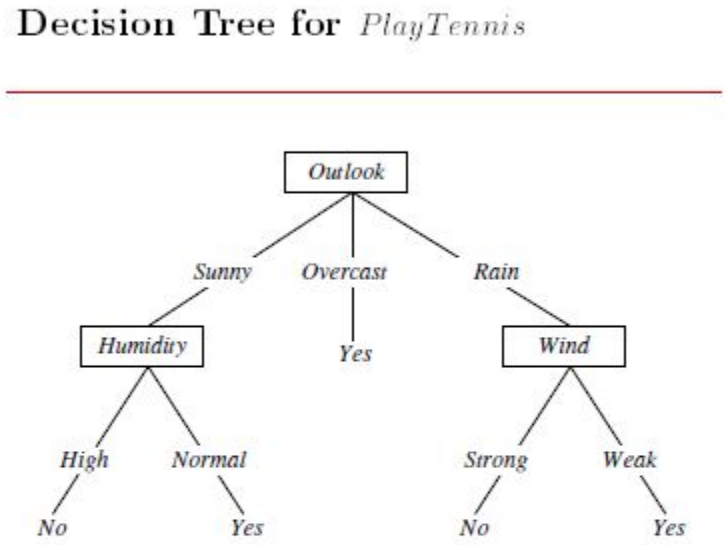

Training Set
Learning Algorithm
Model
Testing Phase
Validation set
Apply Model
Error ??

Random Forest Classifier

- Ensemble of decision trees
- Decision Trees
  - Simple means of inducing rules
    - if (Age is x) and (income is y) then sanction loan

Sample Decision Tree

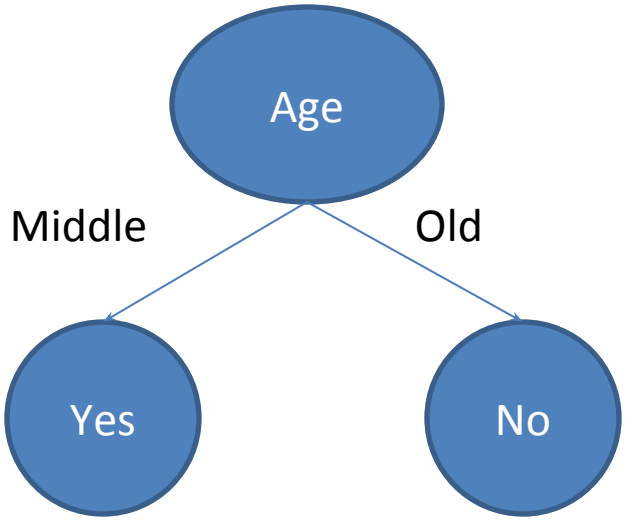
Day	Outlook	Temperature	Humidity	Wind	PlayTenni
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



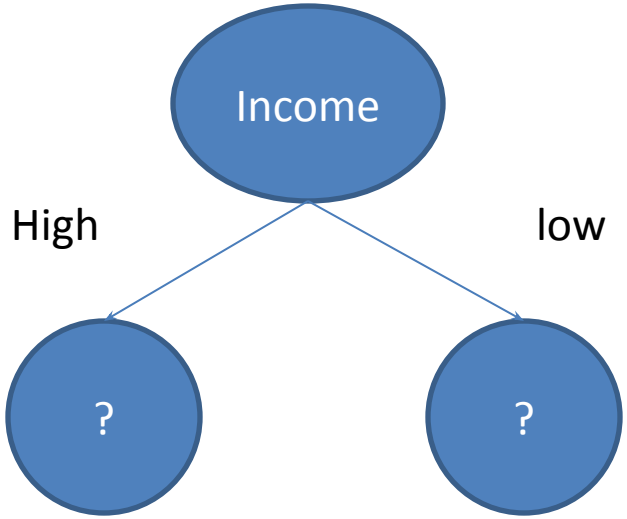
How attributes are selected ?

- Using Metrics
  - Information Gain
  - Entropy

Age	Income	Label
Middle	High	Yes
Middle	Low	Yes
Old	High	No
Old	Low	No



Entropy = 0



Entropy = 1

Random Forest classifier

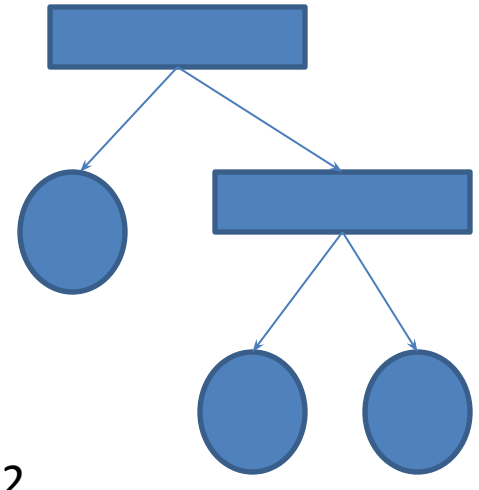
- Problem with decision tree
  - Overfits the training data

Day	Outlook	Temperature	Humidity	Wind	PlayTenni
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

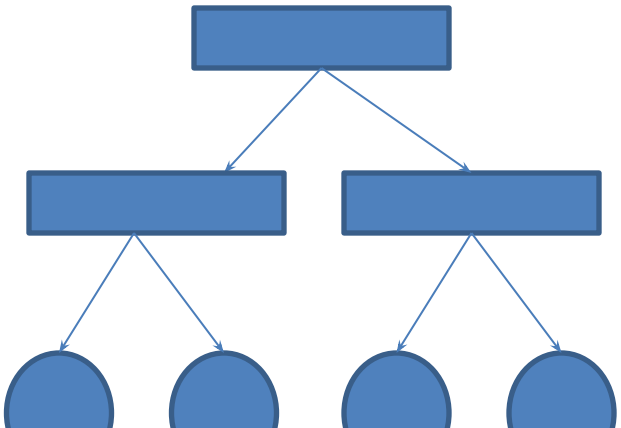
Outlook	Temperature	Play_Tennis
Tuple1		
Tuple 4		
Tuple 8		

Temperature	Wind	Play_Tennis
Tuple 2		
Tuple 5		
Tuple 8		

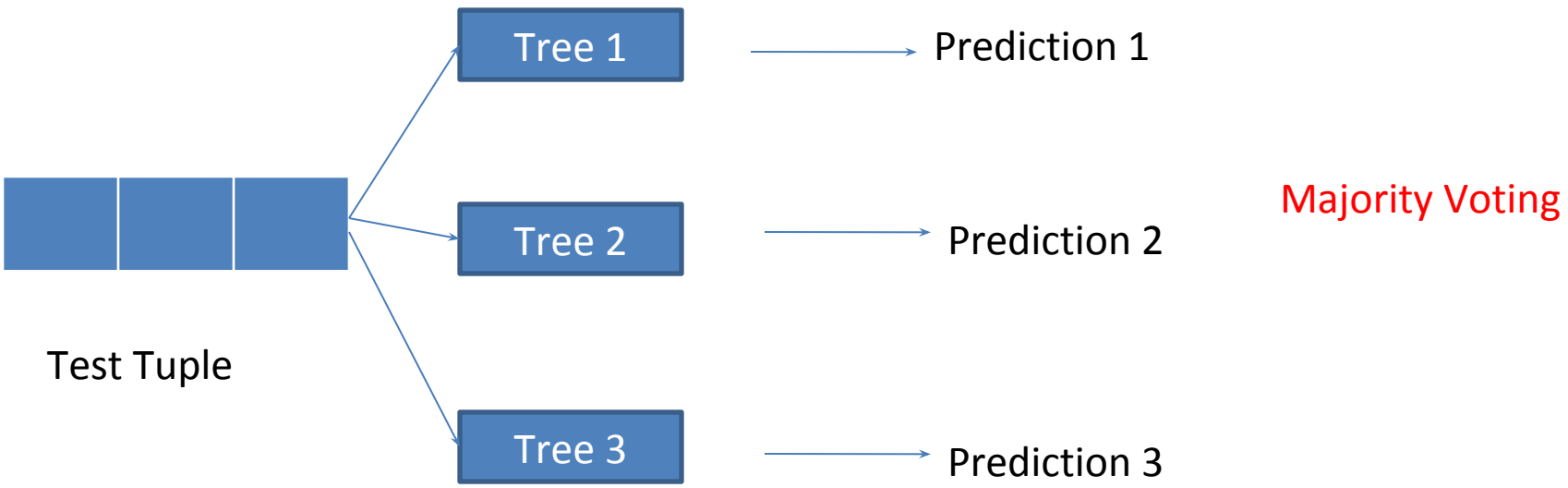
Tree 1



Tree 2



Train & Test Model



Data

- Mllib supports different datatypes (local)
  - Vectors (columnar values)
    - Sparse
    - Dense
  - LabeledPoint
    - associate label with a vector
    - label field ---- > double value
    - features field ---- > Vector

```
from pyspark.mllib.regression import LabeledPoint
pos = LabeledPoint (1.0, [1.0,0.0,3.0])
```

Mllib Package

- Random Forest Classifier
  - pyspark.mllib.tree
  - import RandomForest, RandomForestModel
- Train
  - model = RandomForest.trainClassifier (Data,
    numClasses =2,
    CategoricalFeatureInfo{ Map(0->2,4>10)},
    featureSubsetStrategy =“auto”,
    impurity=“gini”,
    maxDepth=4,
    maxBins=32,
    numTrees=3)

- Problem Specification Parameters
  - Algorithm
  - numClasses
  - categoricalFeaturesInfo
- Stopping Criteria
  - maxDepth
  - minInstancesPerNode
  - minInfoGain
- Tunable Parameters
  - Impurity

Testing Model using MLlib

- testdata – dataset with many tuples
- predictions=model.predict(testdata.map(lambda x:x.features)

- labelsAndPredictions=testdata.map(lambda lp:lp.label).zip(predictions)

Evaluation

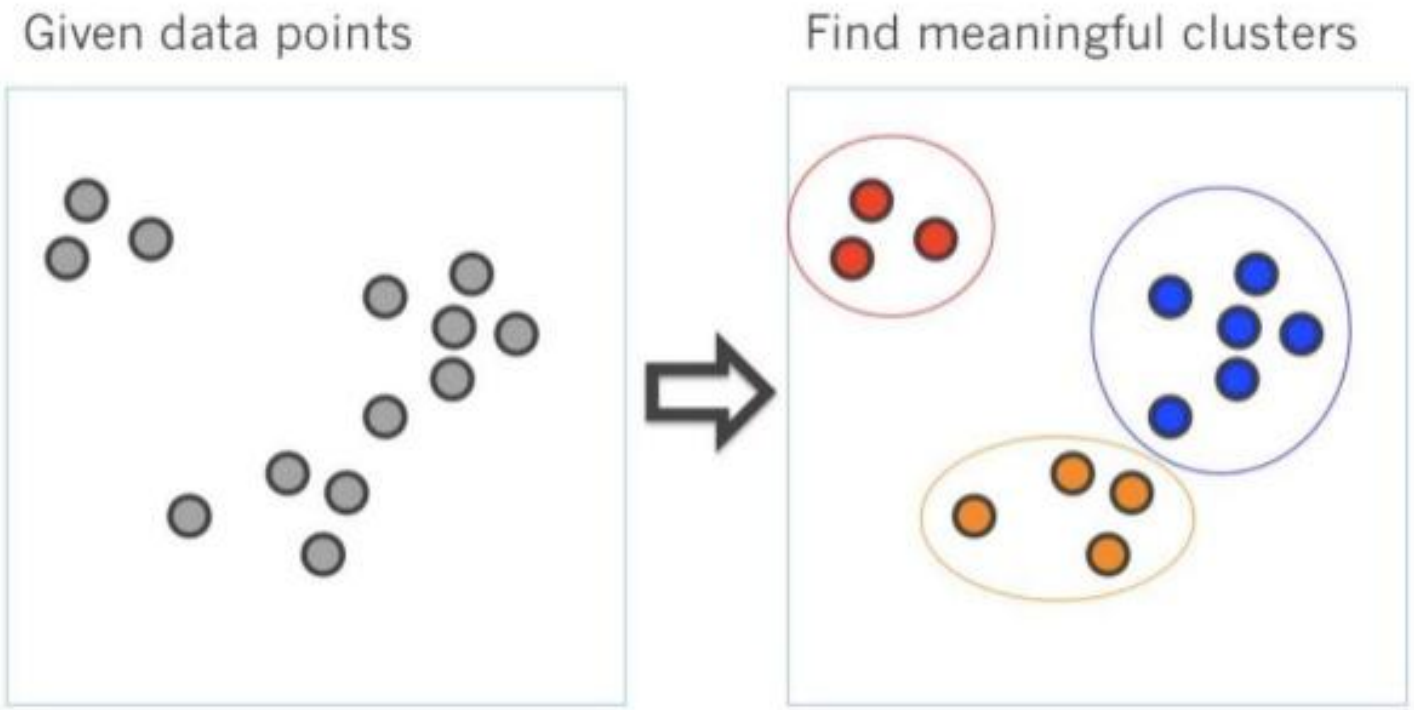
- testErr=labelsAndPredictions.filter(lambda (v,p) : v!=p).count() / float(testData.count())
- print('Test Error = ' + str(testErr))

- model.save(sc,"MyModelPath")
- sameModel = RandomForestModel.load(sc,"MyModelPath")

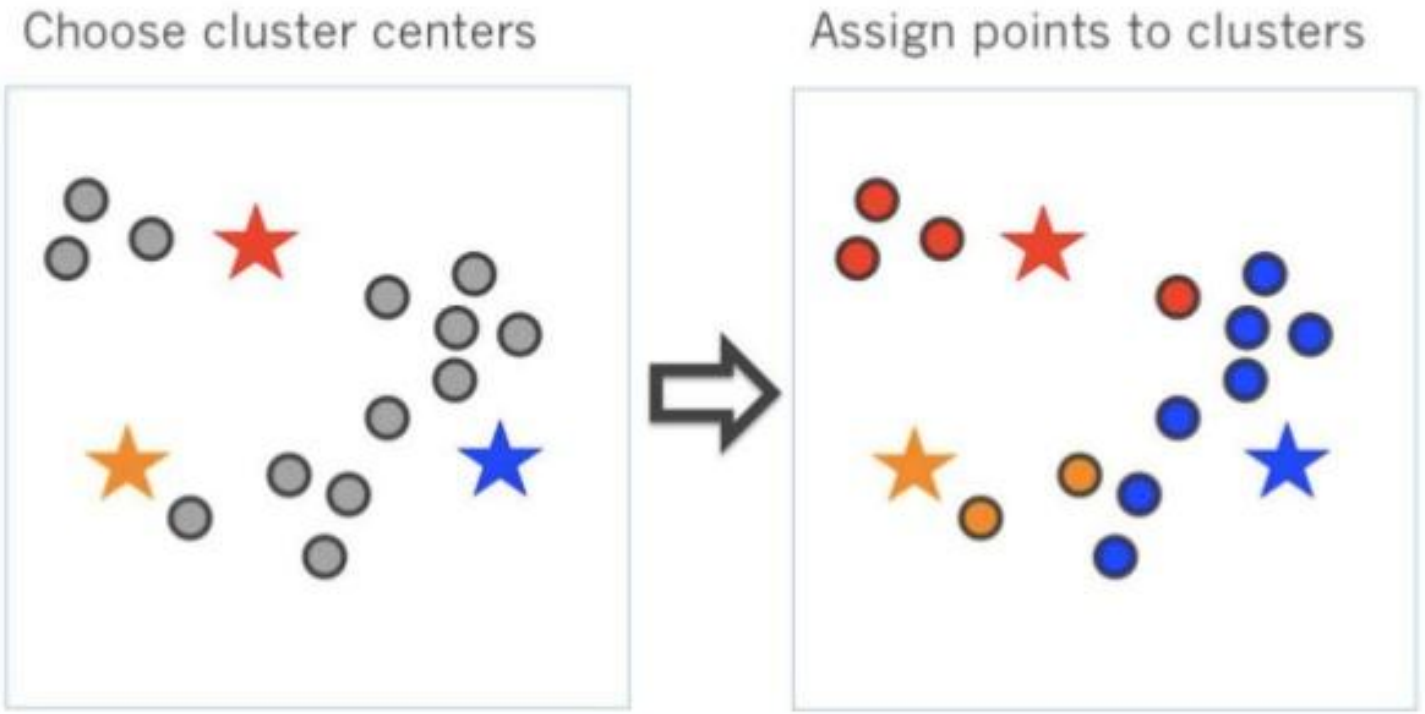
K-means Clustering

- Most Commonly used Clustering algorithm
- partition n observations into k clusters in which each observation belongs to the cluster with nearest mean, serves as the model of the cluster
- It works only with numerical attributes

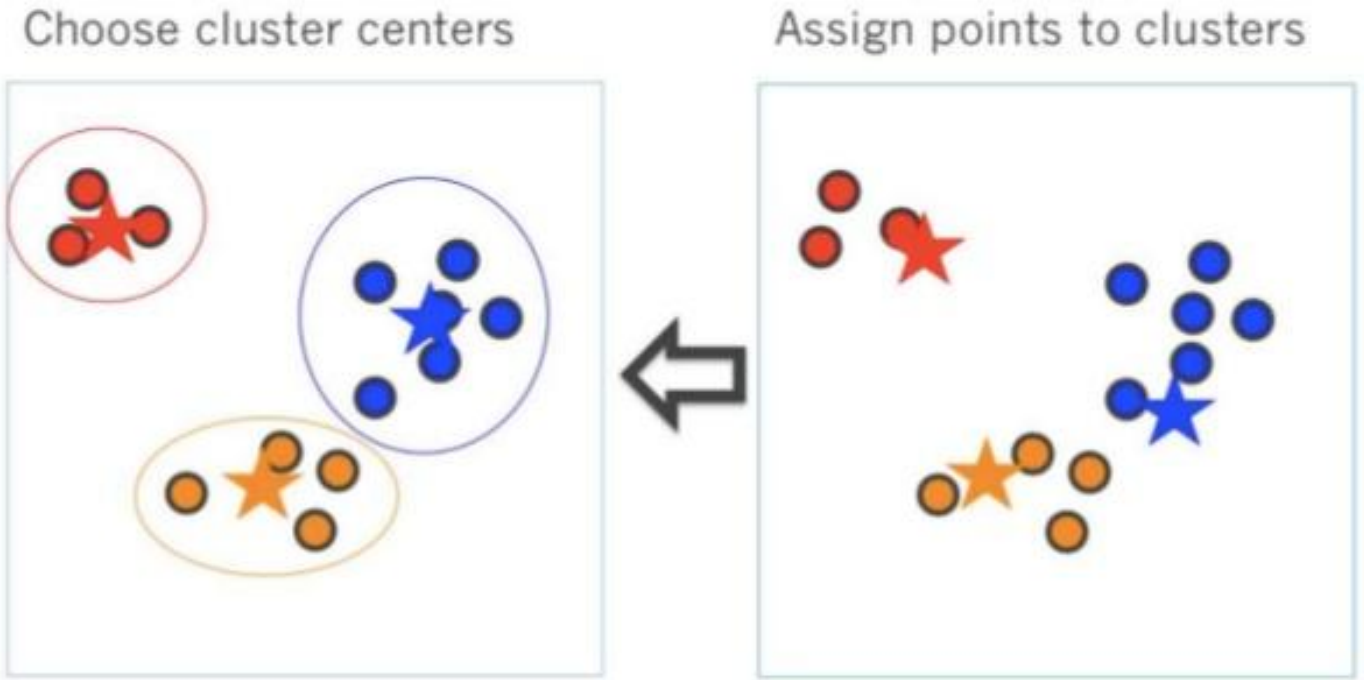
Clustering with K-Means



Clustering with K-Means



Clustering with K-Means



Algorithm

- Runs iteratively
- Starts with initializing number of clusters and the cluster center
  - Assign data points to clusters
  - Recompute cluster centre
  - Repeat steps till error metric is reduced to a threshold

Mllib package- clustering

- Kmeans input data:
  - Vector
  - To create vectors , numpy package could be used

```
from numpy import array

data = sc.textFile("data/mllib/kmeans_data.txt")
parsedData = data.map(lambda line: array([float(x)
   for x in line.split(' ')]))
```

- Train Model :

```
clusters = KMeans.train(
                parsedData, 2,
                maxIterations=10,
              runs=10,
   initializationMode="random")
```

- Test Model

```
def error(point):
center =
clusters.centers[clusters.predict(point)]
return sqrt(sum([x**2 for x in (point -
center)]))
```

```
WSSSE = parsedData.map(lambda point: error(point)).reduce(lambda x, y: x + y)
print("Within Set Sum of Squared Error = " + str(WSSSE))
```

- # Save and load model

```
clusters.save(sc, "myModelPath")
sameModel = KMeansModel.load(sc, "myModelPath")
```

MLlib

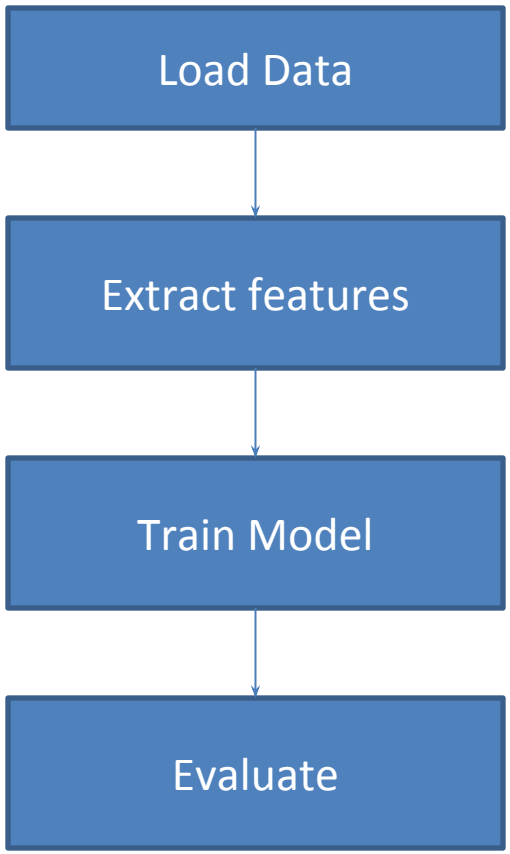
- Machine learning has to be easy and scalable
  - Capable of learning from large datasets.
  - Easy way to build machine learning applications.

Machine Learning Workflow

- Scalable -- Expandable (it should work even if the data grows enormously)
- Machine learning pipeline components
  - Feature Extraction
  - Supervised Learning
  - Model Evaluation
  - Exploratory data analysis

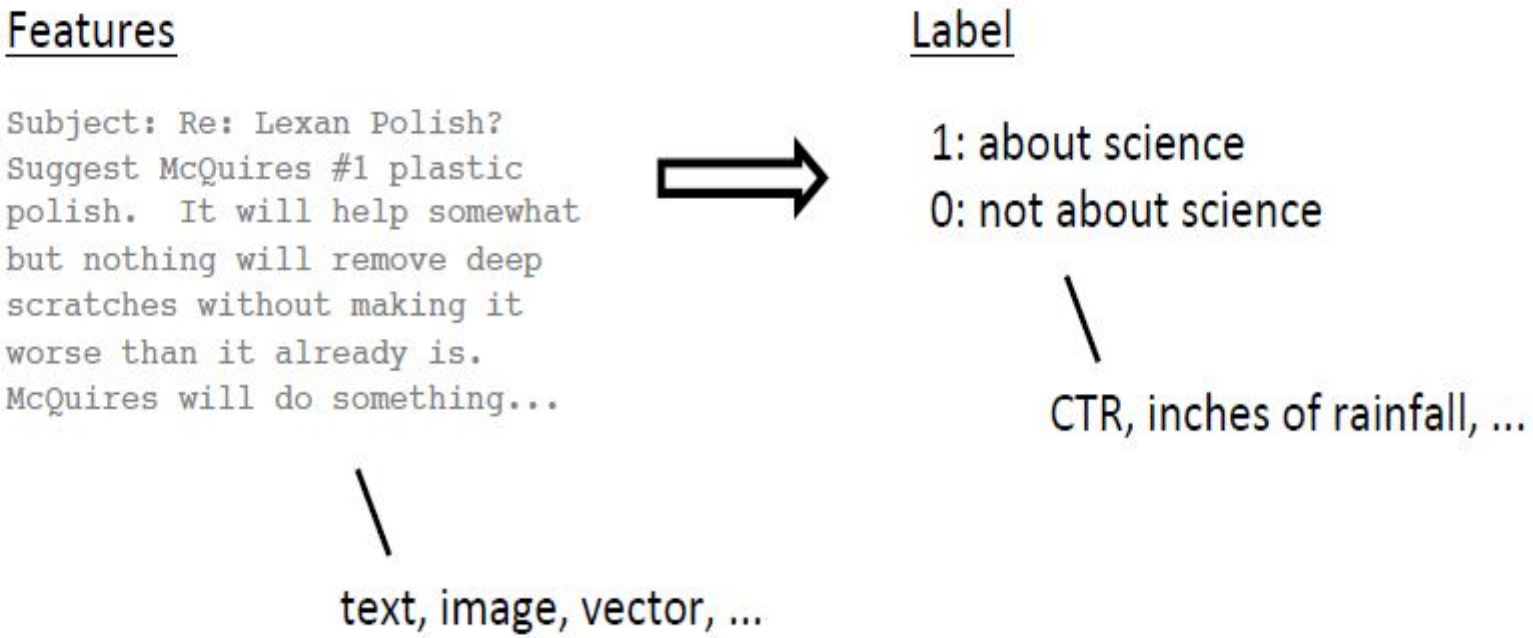
ML workflow

- Why do we need ML workflow



Text Classification

- Given text predict its topic



Training & Testing

Training

Given labeled data:

RDD of (features, label)

```
Subject: Re: Lexan Polish?
Suggest McQuires #1 plastic
polish.  It will help...
```
Label 0

```
Subject: RIPEM FAQ
RIPEM is a program which
performs Privacy Enhanced...
```
Label 1

...

Learn a model.

Testing/Production

Given new unlabeled data:

RDD of features

```
Subject: Apollo Training
The Apollo astronauts also
trained at (in) Meteor...
```

```
Subject: A demo of Nonsense
How can you lie about
something that no one...
```

Use model to make predictions.

Training & Testing

Training

Given labeled data:

RDD of (features, label)

```
Subject: Re: Lexan Polish?
Suggest McQuires #1 plastic
polish.  It will help...
```
Label 0

```
Subject: RIPEM FAQ
RIPEM is a program which
performs Privacy Enhanced...
```
Label 1

...

Learn a model.

Testing/Production

Given new unlabeled data:

RDD of features

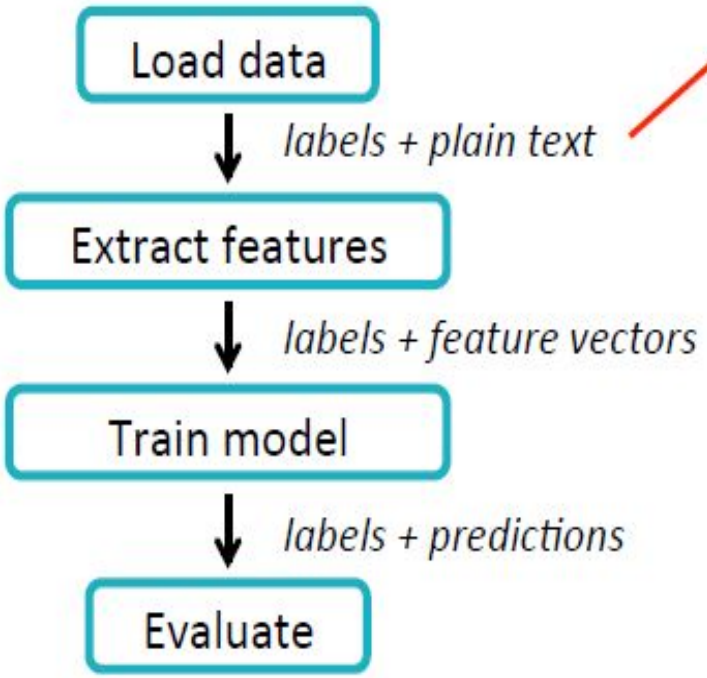
```
Subject: Apollo Training
The Apollo astronauts also
trained at (in) Meteor...
```
→ Label 1

```
Subject: A demo of Nonsense
How can you lie about
something that no one...
```
→ Label 0

Use model to make predictions.

Example ML Workflow

Training



Problems with Mllib

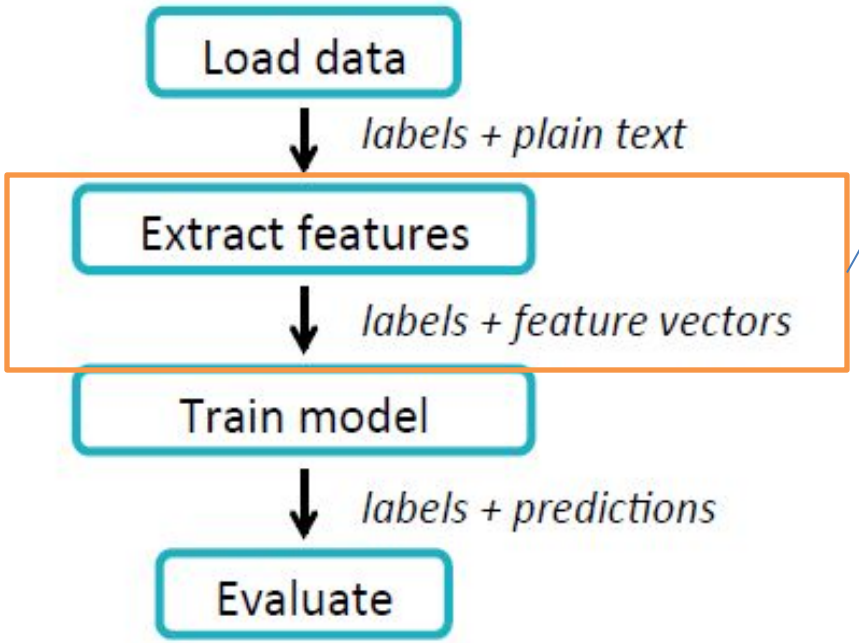
Create many RDDs

retrieve words from text
convert it to features (new)

Explicit Zip
associate data point with features

Example ML Workflow

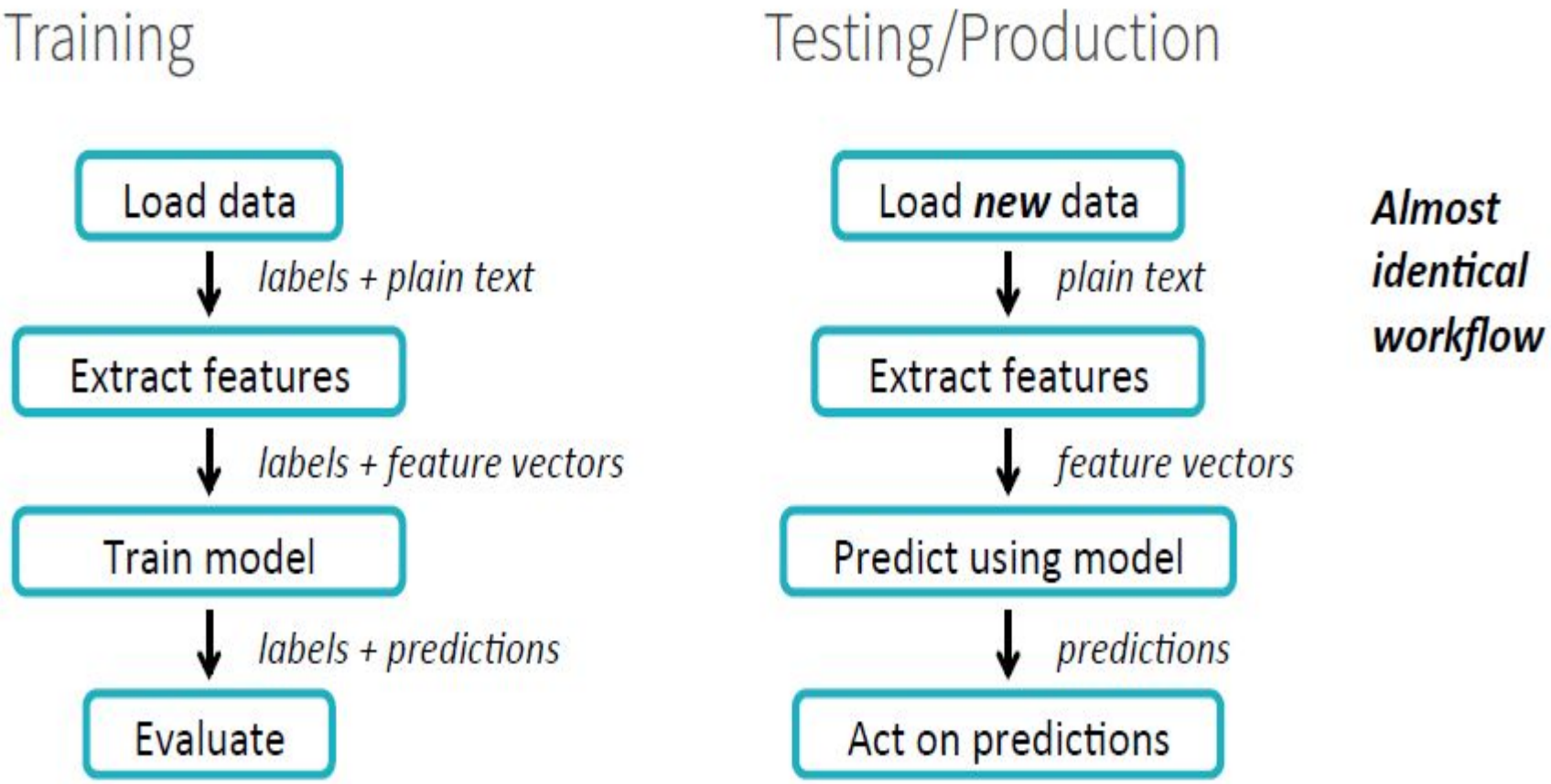
Training



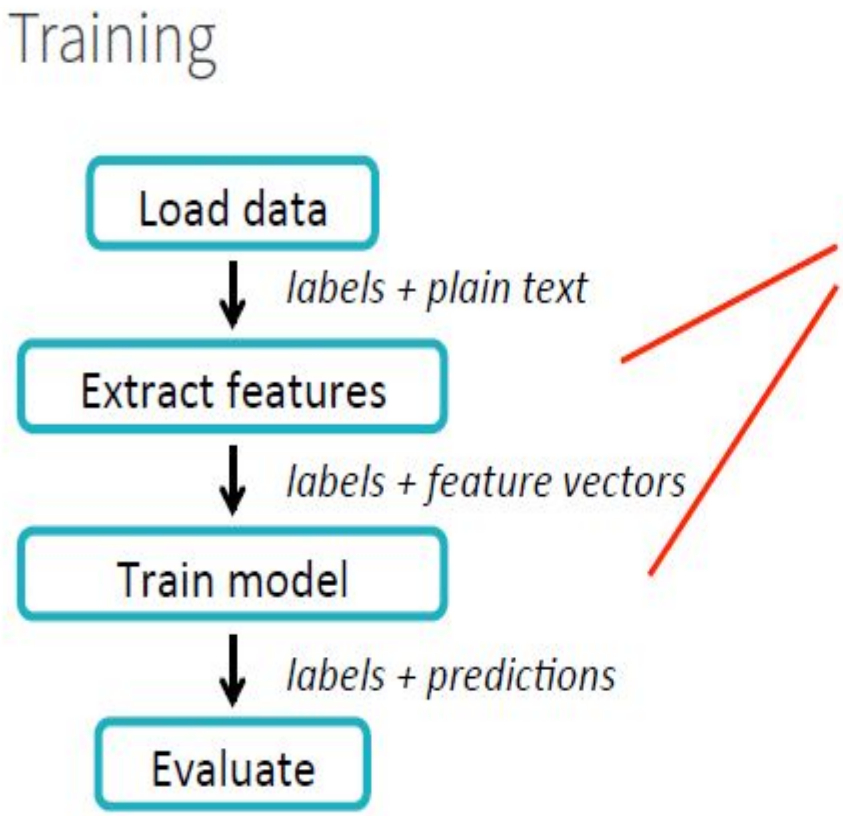
Write as a script

- Not modular
- Difficult to re-use workflow

Example ML Workflow



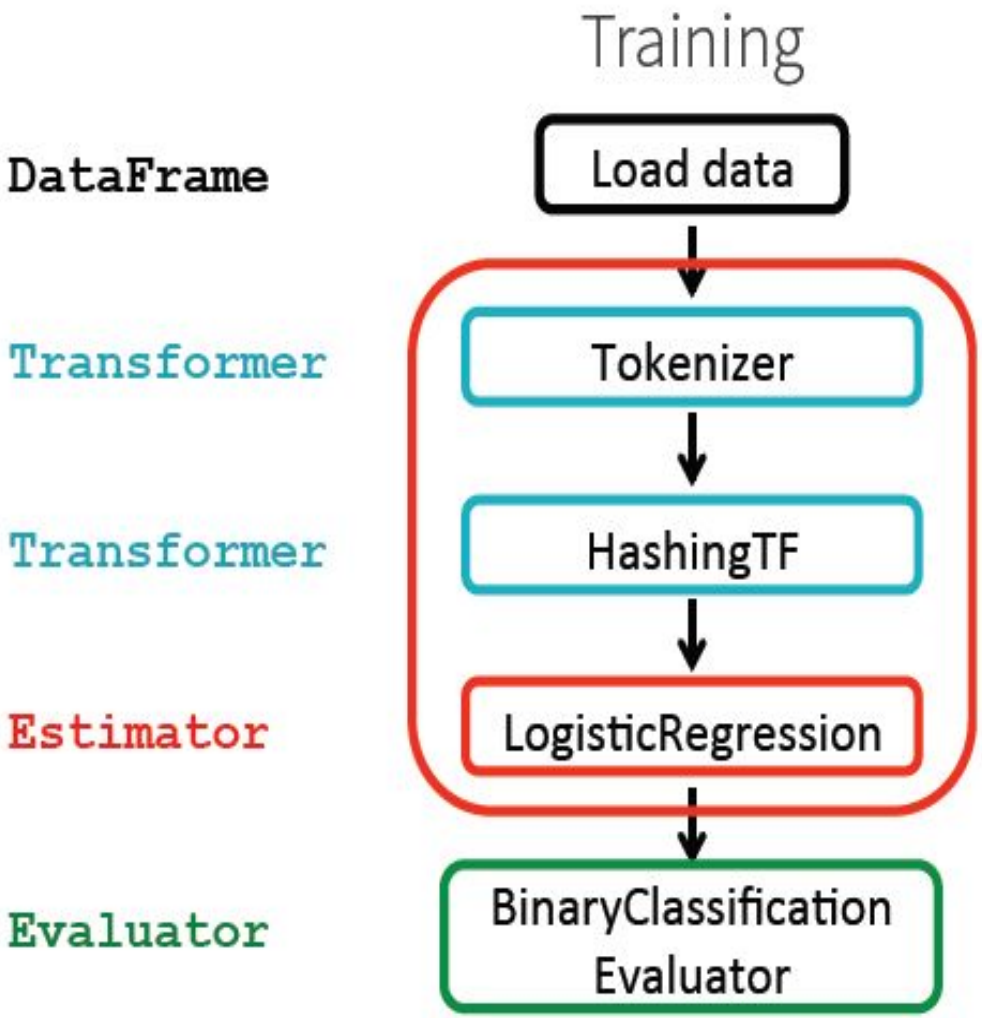
Example ML Workflow

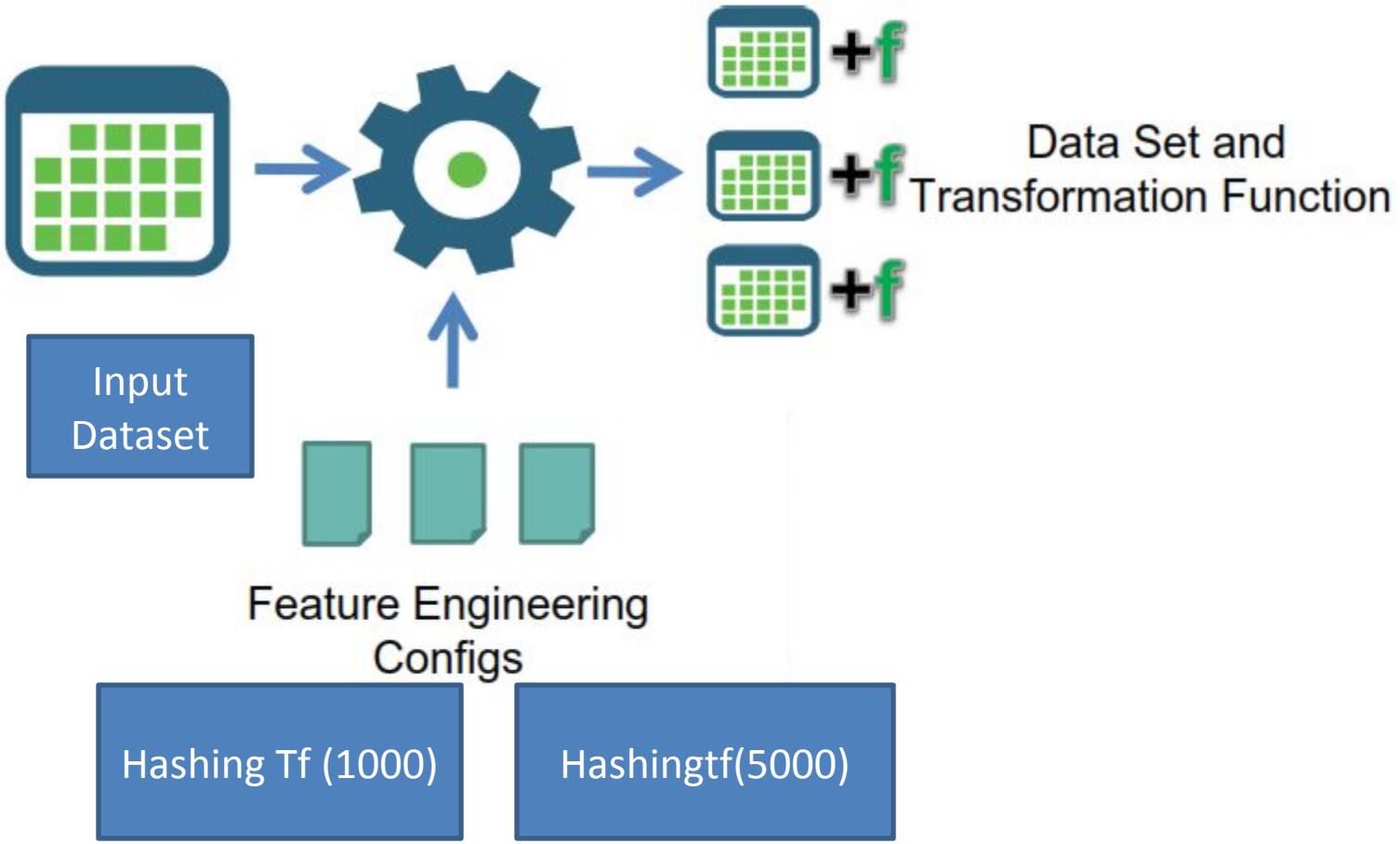


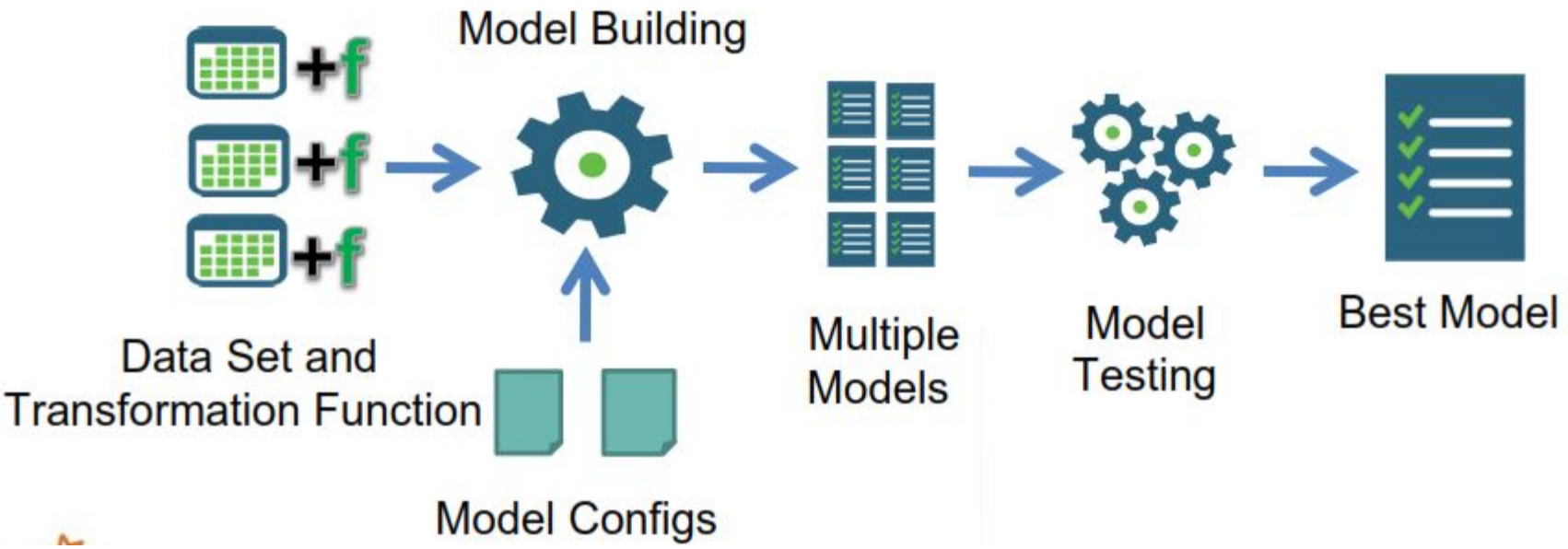
Next Biggest Challenge

Parameter tuning

- Key part of ML
- Involves training many models
  - For different splits of the data
  - For different sets of parameters


Training
DataFrame
Load data
Transformer
Tokenizer
Transformer
HashingTF
Estimator
LogisticRegression
Evaluator
BinaryClassification
Evaluator


+f
+f
+f
Data Set and
Transformation Function
Input
Dataset
Feature Engineering
Configs
Hashing Tf (1000)
Hashingtf(5000)


Model Building
Data Set and
Transformation Function
Model Configs
Multiple
Models
Model
Testing
Best Model

References

Books :

- **Learning Spark: Lightning-Fast Big Data Analysis by** Holden Karau, Andy Konwinski, Patrick Wendell & Matei Zaharia

Online Courses :

edx course - **CS120x Distributed Machine Learning with Apache Spark**