Introduction to Machine Learning on Apache Spark

02 | Using ML Pipelines

Microsoft

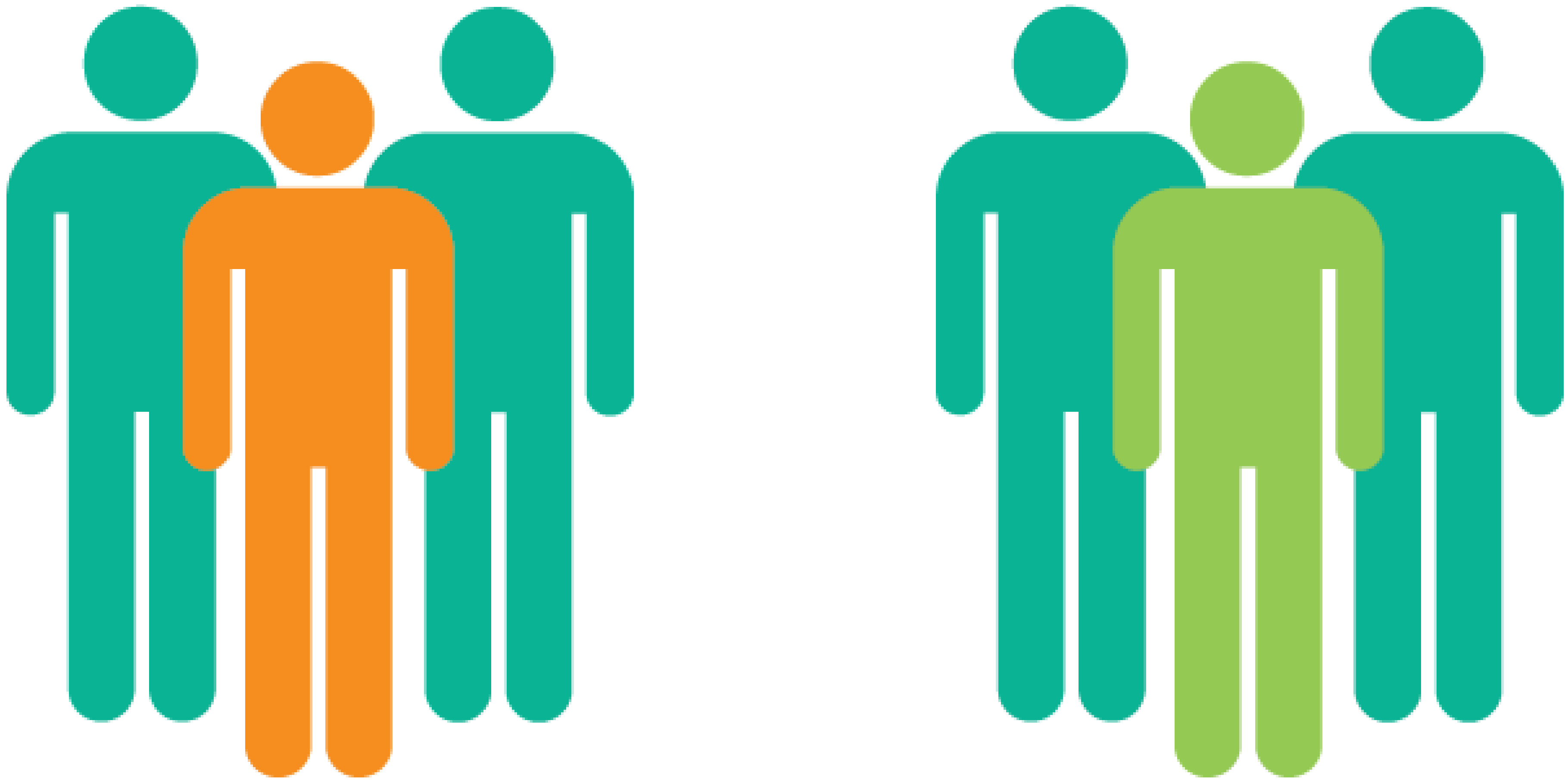
Richard Conway | Microsoft Azure MVP, Elastacloud

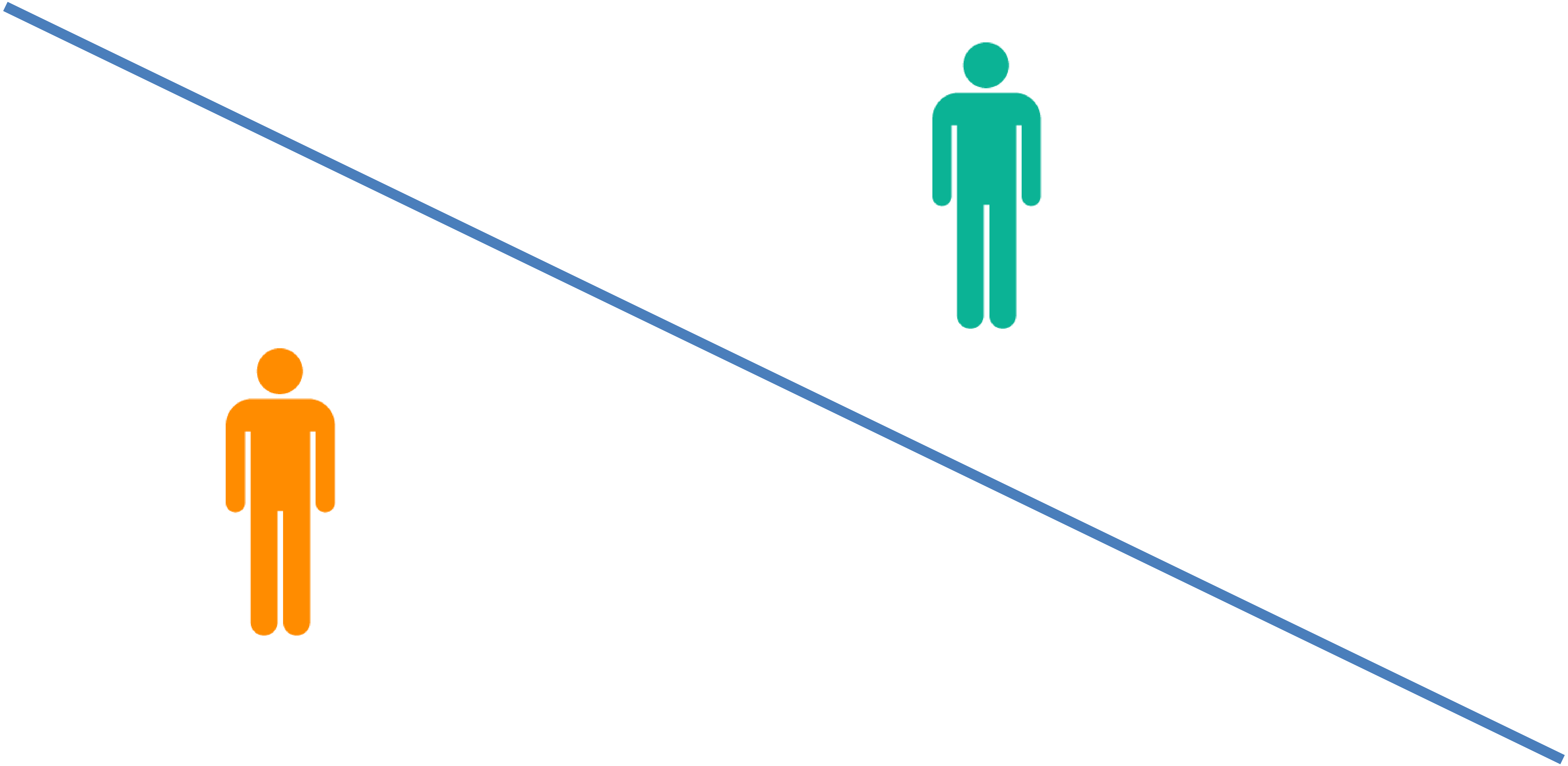
- What is Binary Classification? What is Multiclass Classification?
- What is regression?
- What is collaborative filtering?
- What is Unsupervised Learning?
- What is K-Means Clustering?
- How do I use Spark MLLib?
- How to I build Spark ML programs?
- How can I build a workflow in ML?
- What is a pipeline?
- What is a Transformer?
- What is an Estimator?

What is Binary Classification?

- Also called Binomial Classification
- Take a dataset and classify contents into two groups

- medical testing to determine if a patient has certain disease or not – the classification property is the presence of the disease;
- A "pass or fail" test method or quality control in factories; i.e. deciding if a specification has or has not been met: a Go/no go classification.
- An item may have a qualitative property; it does or does not have a specified characteristic
- information retrieval, namely deciding whether a page or an article should be in the result set of a search or not – the classification property is the relevance of the article, or the usefulness to the user.





- Decision Trees
- Random forests
- Bayesian networks
- Support vector machines
- Neural networks
- Logistic regression

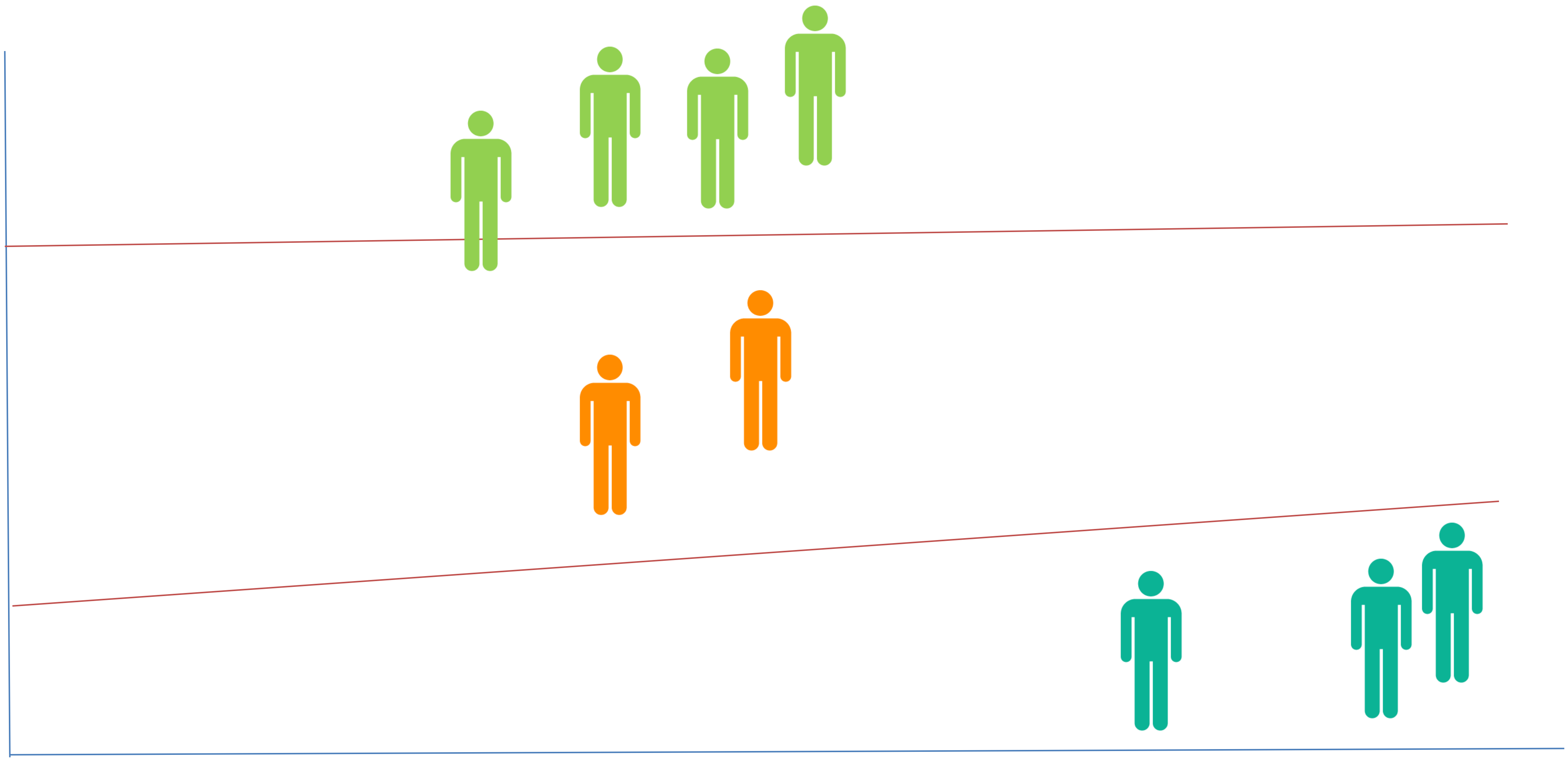
```
val categoricalFeaturesInfo = Map[Int, Int]()
val impurity = "gini"

val model = DecisionTree.trainClassifier(trainingData, numClasses,
categoricalFeaturesInfo, impurity, maxDepth, maxBins)
```

DEMO

Building a Binary Classification Model

What is Multiclass Classification?



```
val allData = data.randomSplit(Array(0.7, 0.3), seed = 11L)
val (training, test) = (allData(0), allData(1))
val model = new
LogisticRegressionWithLBFGS().setNumClasses(3).run(training)

val predictionAndLabels = test.map { case LabeledPoint(label,
features) =>
  val prediction = model.predict(features)
  (prediction, label)
}
```

DEMO

Building a Multiclass Classification Model

How do I measure the success of a classifier?

predicted

actual

	Right	Left	Moderate
Right	3	2	0
Left	4	12	2
Moderate	2	1	4

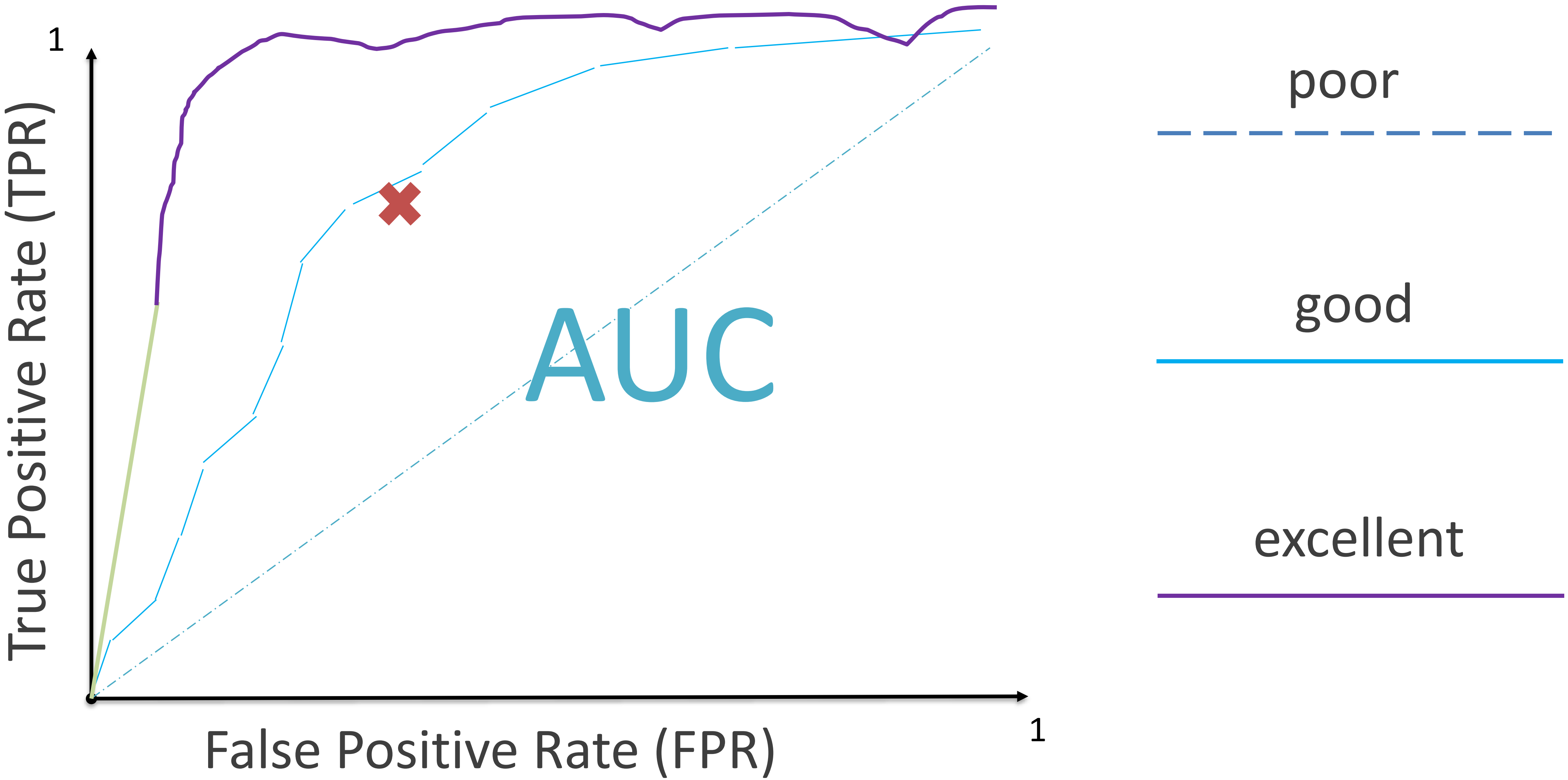
for Left

Accuracy

12 true positives	3 false positives	15
6 false negatives	9 true negatives	15
18	12	

Recall

Specificity

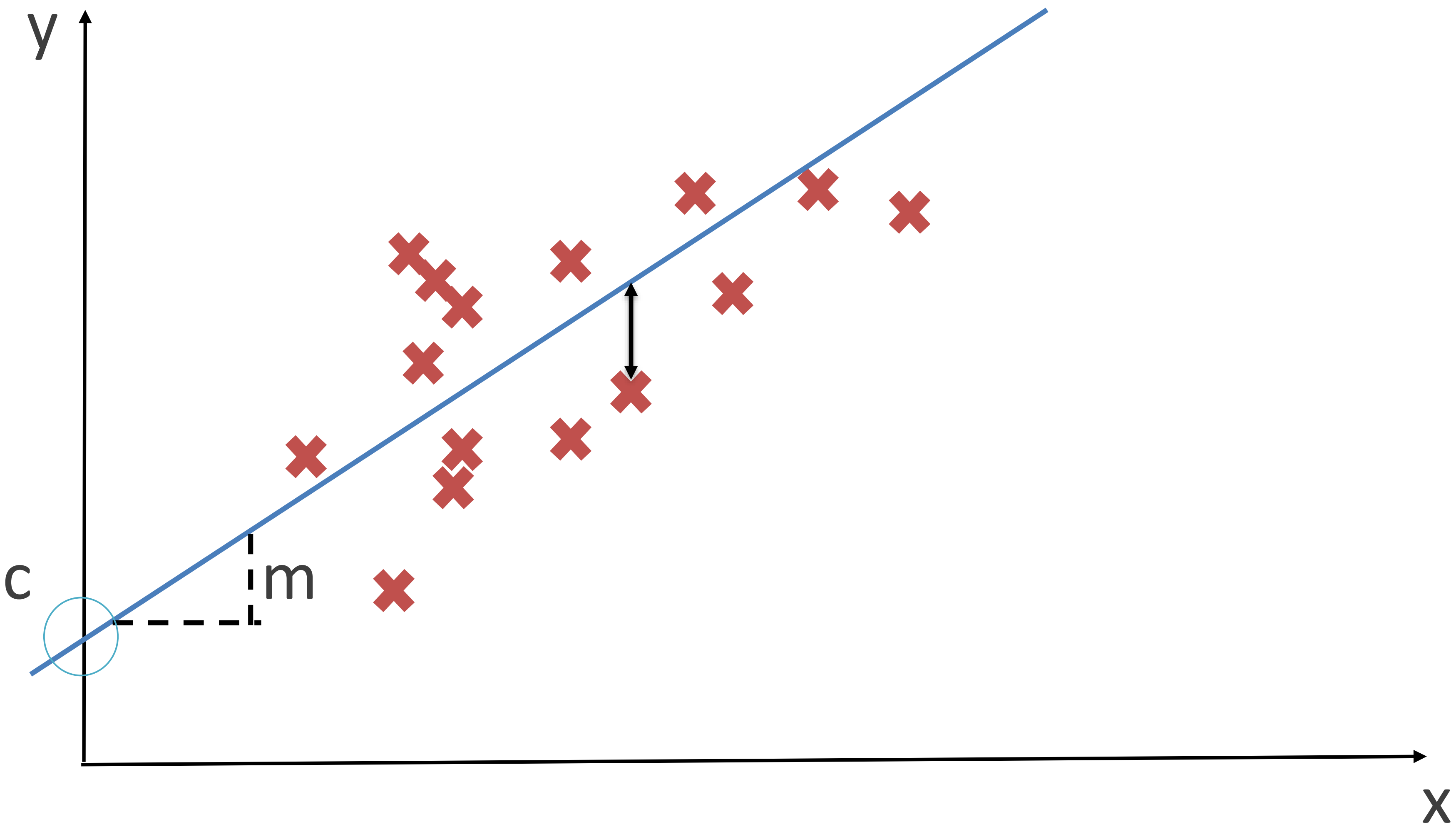

1
True Positive Rate (TPR)
AUC
1
False Positive Rate (FPR)
poor
good
excellent

```
val metrics = new BinaryClassificationMetrics(labelAndPreds)

//show the area under the curve
val roc = metrics.roc
val auROC = metrics.areaUnderROC

//precision recall curve
 val PR = metrics.pr
 val auPR = metrics.areaUnderPR
```

What is regression?



$y = mx + c$

- Machine Learning uses Linear Regression and can process multiple features
- Assuming we have temperature, wind speed and rainfall and use this to predict number of sales
- We derive a “cost function” which is used in conjunction with the feature vector
- We can apply Stochastic Gradient Descent to iteratively find the best fit for the line

- Least square linear regression (LR)
- Decision trees (TREE) Boosting trees (BOOST)
- Neural networks (NN)

```
val numIterations = 600
val stepSize = 0.1
val algorithm = new LinearRegressionWithSGD()
      .setIntercept(true)
algorithm.optimizer.setNumIterations(numIterations)
algorithm.optimizer.setStepSize(stepSize)

val model = algorithm.run(scaledData)
```

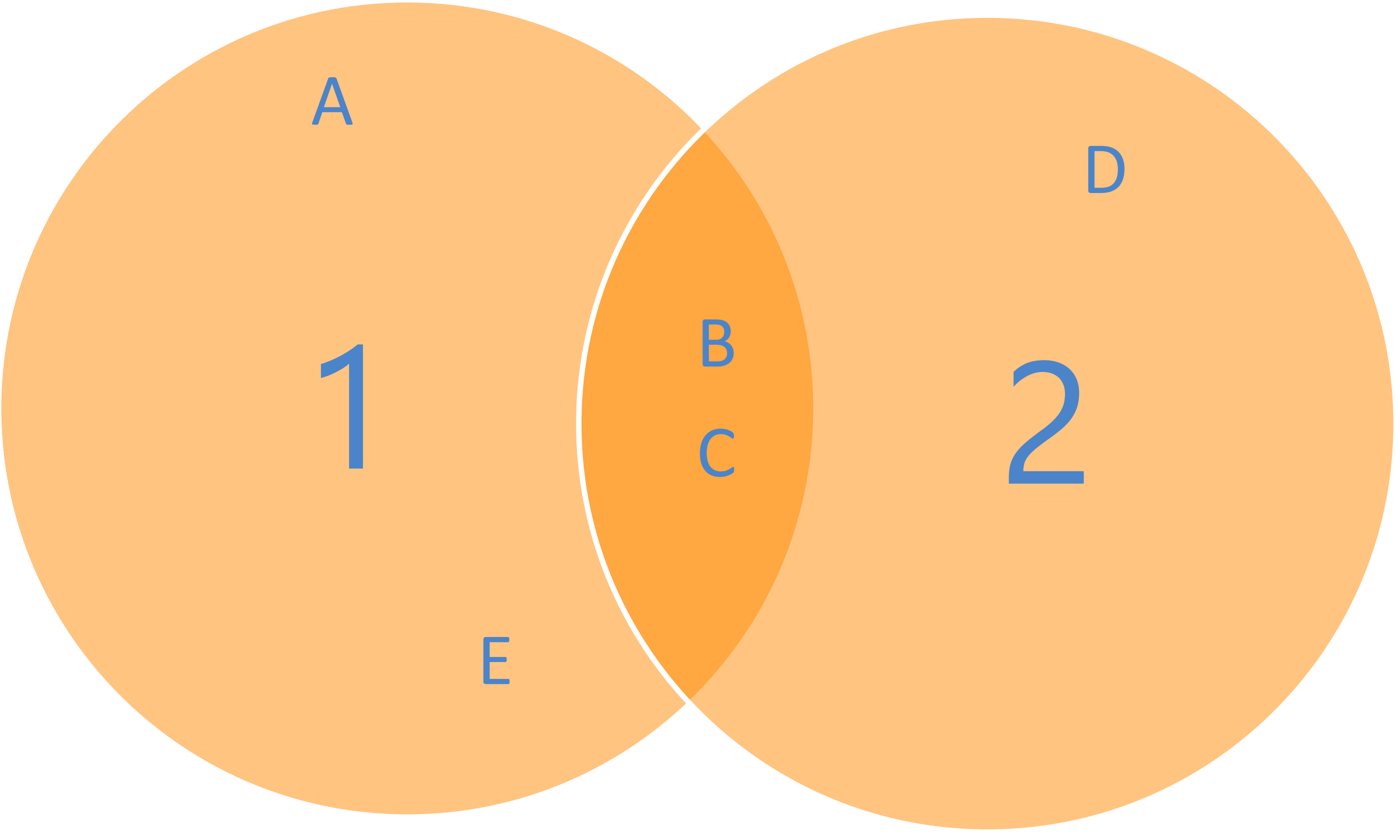
What is collaborative filtering?

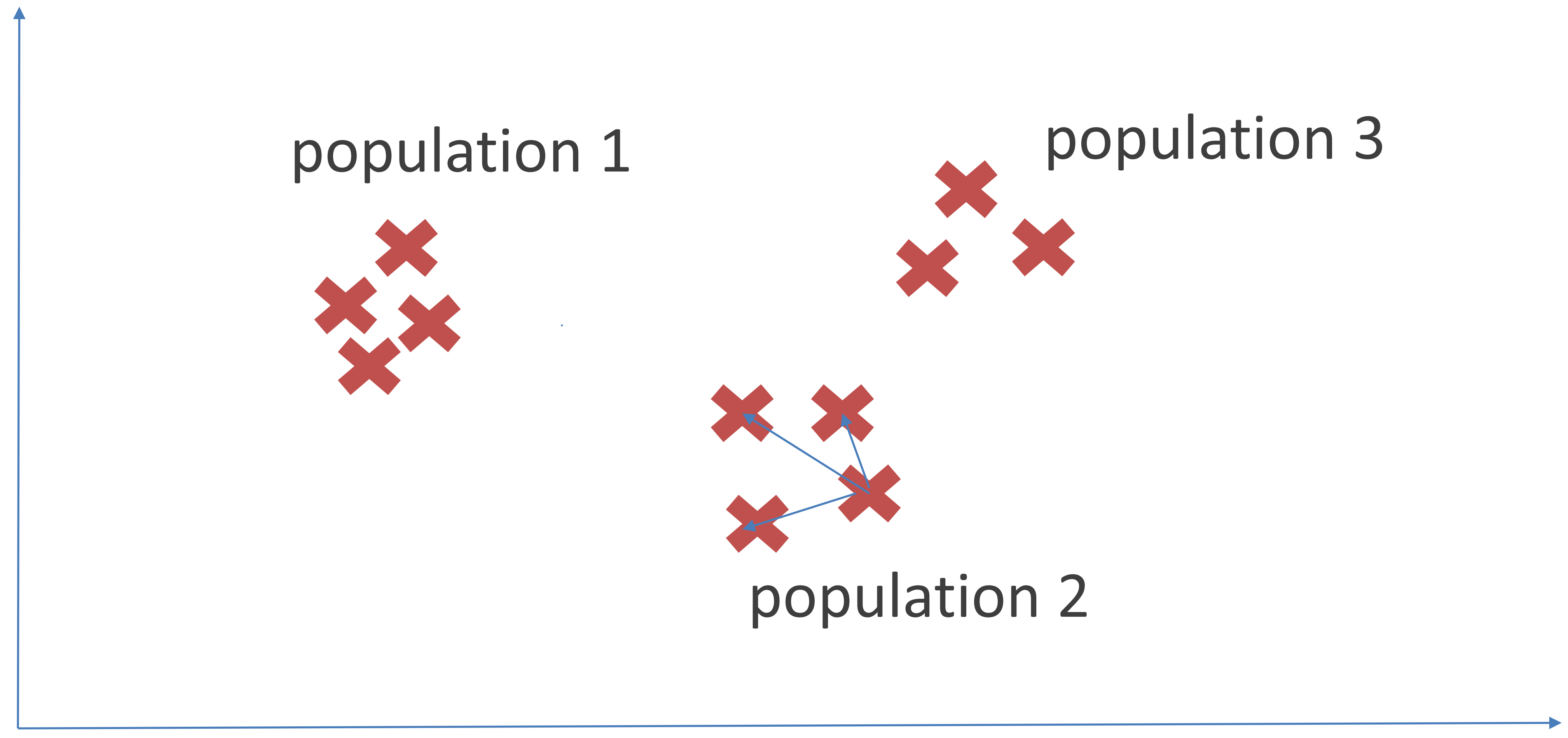
- Recommender systems commonly use *Collaborative Filtering*
- Collaborative filtering can use *ratings* given by users
- Collaborative filtering uses a variety of algorithms to determine distance between users
- Rankings done through KNN

- Recommendations can be *implicit* or *explicit*
- Evaluation can take place by calculating the Mean Square Error
- Recommenders can be user-based or item-based

- The “cold start” problem
- Not enough data to rate items
- Gaming the system
- “Grey sheep” users
- Invalid references
- Product bias
- Scalability

	1	2	3	4
1	A,C	A	A,C	A
2		A,B	A	-
3			A,C	A
4				D,E


A
D
1
B
C
2
E


population 1
population 3
population 2

- Pearson Correlation
- Euclidean Distance
- Cosine Similarity
- Spearman Correlation
- Tanimoto Coefficient
- LogLikelihood Coefficient

```
val factorization = ALS.trainImplicit(trainingData, 9, 10)

//get recommended movies for a particular person/user
val recommended = factorization.recommendProducts(656,14)
```

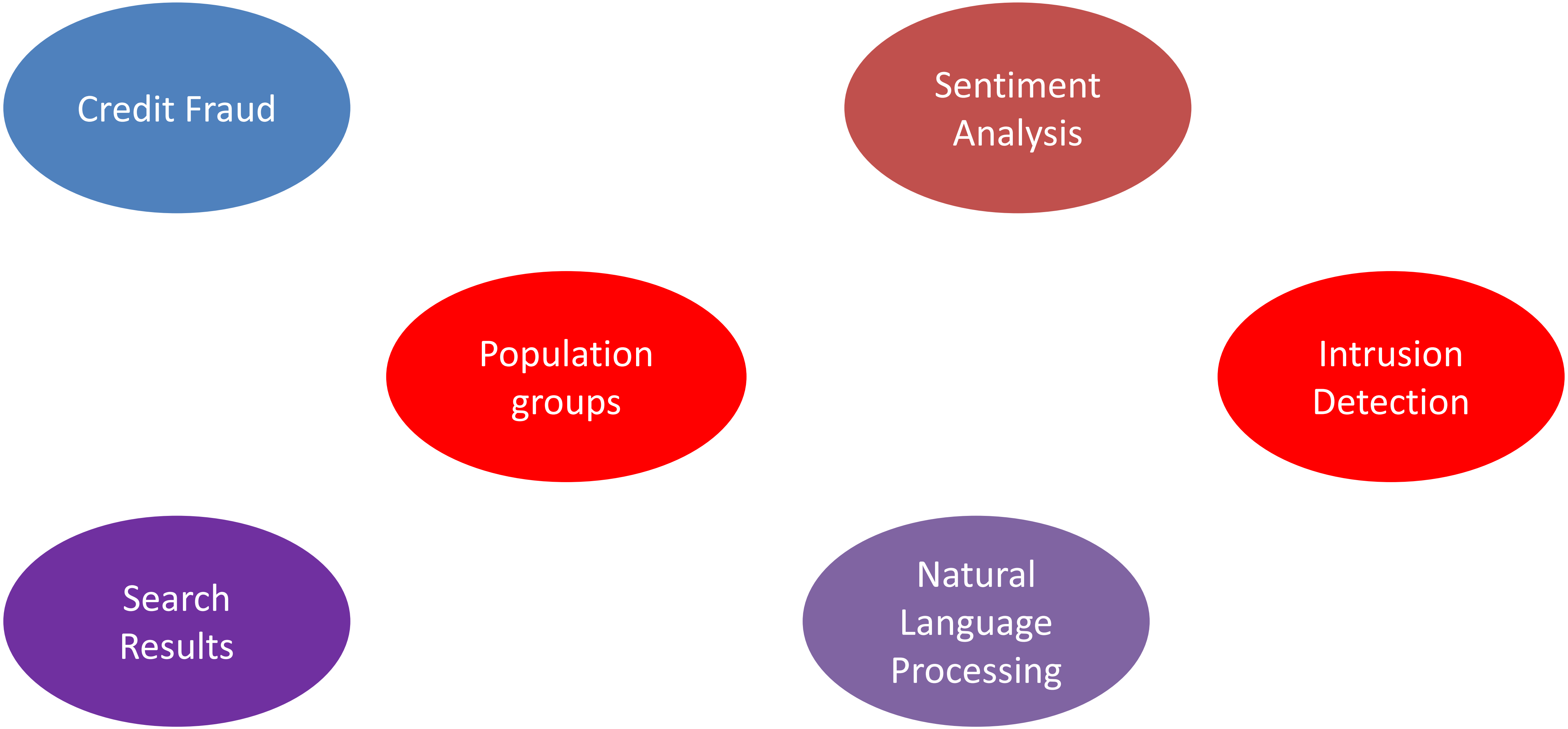
What is Unsupervised Learning?

Unsupervised learning looks for hidden structure in unlabelled data.

In the supervised learning we have labels and we can check a signal for any errors.

There is no signal in unsupervised learning so no means to evaluate success only the emergent structure which becomes apparent.

What can I use it for?



Credit Fraud

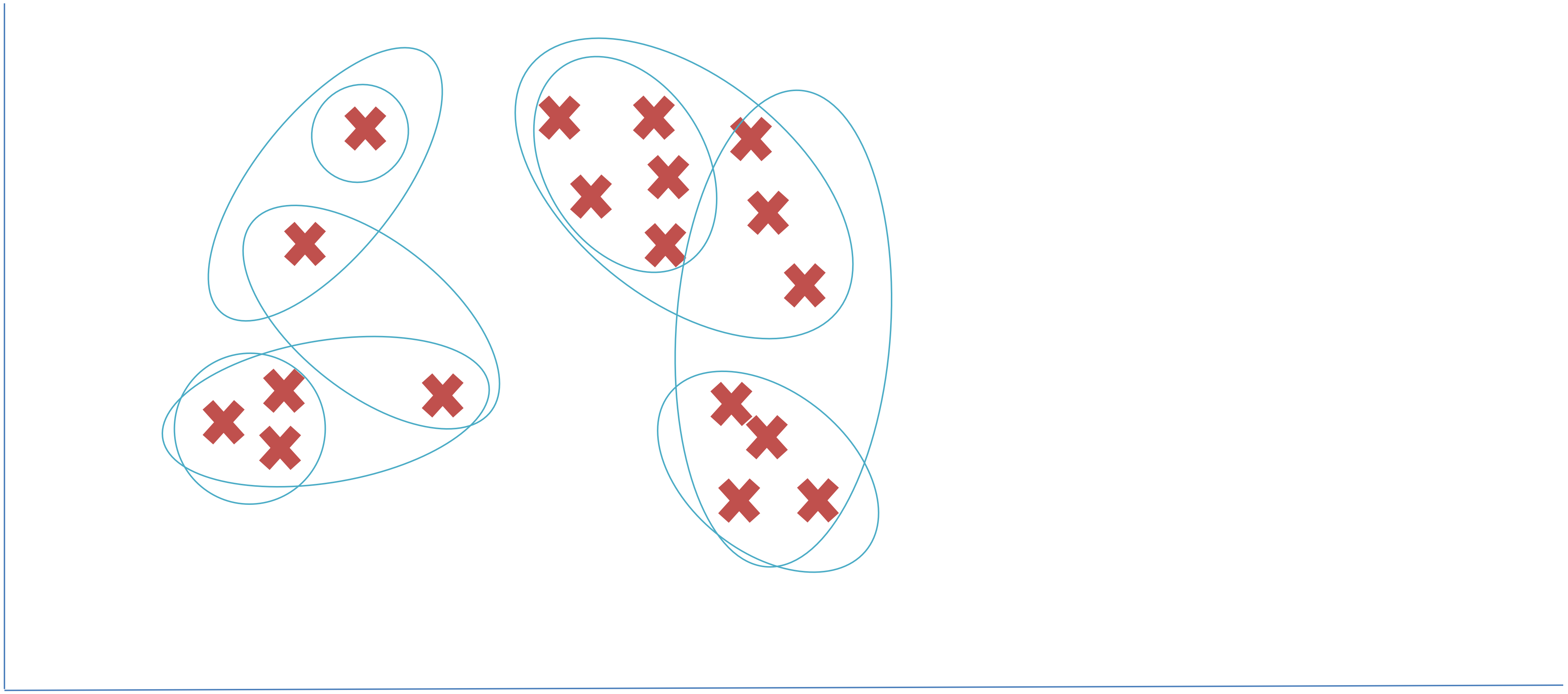
Sentiment Analysis

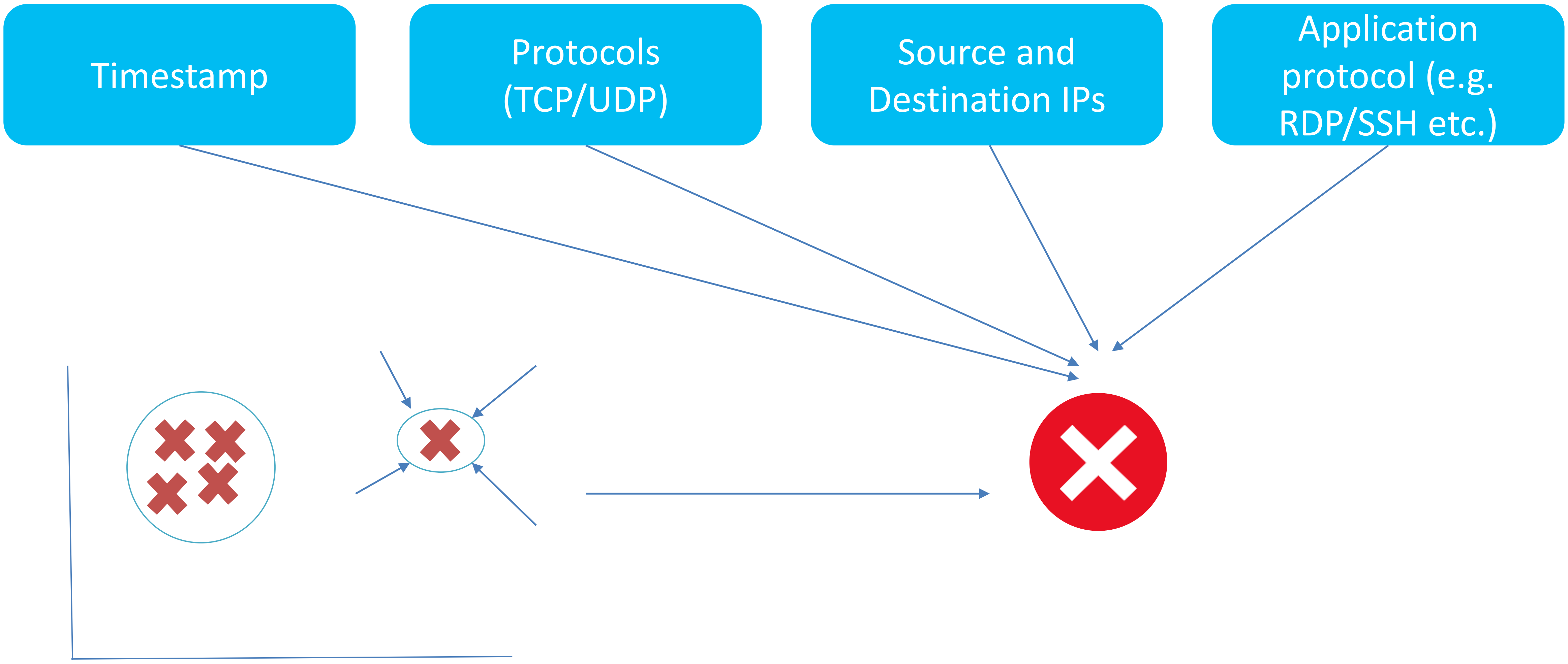
Population groups

Intrusion Detection

Search Results

Natural Language Processing





Timestamp

Protocols (TCP/UDP)

Source and Destination IPs

Application protocol (e.g. RDP/SSH etc.)

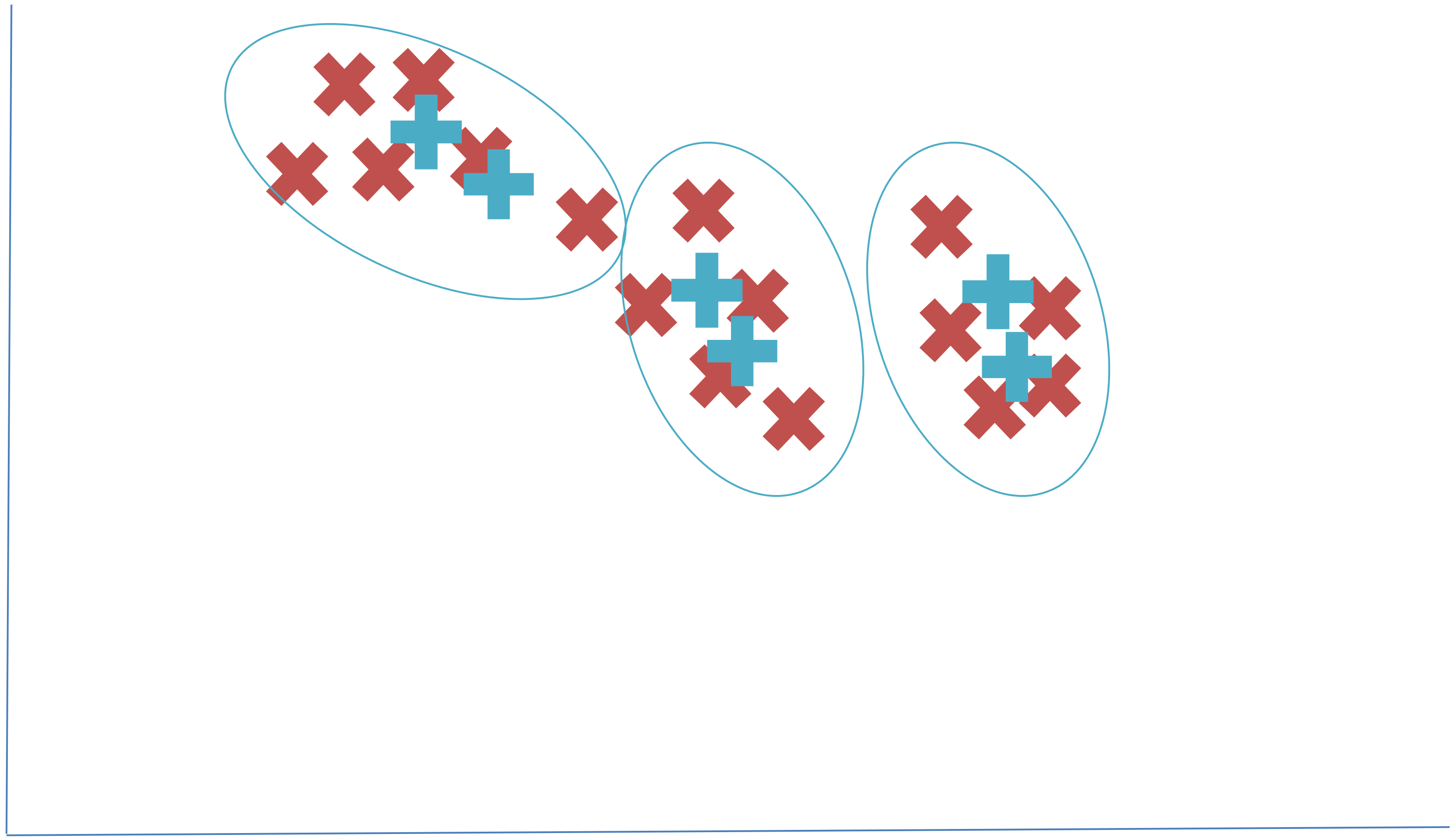
- Density Estimates
- Principal Component Analysis (PCA)
- Singular Value Decomposition (SVD)
- Self Organizing Map (SOM)
- Adaptive Resonance Theory (ART)

```
parsedData = input.map(lambda line: array([float(x) for x in
line.split(',')[0:4]]))

from pyspark.mllib.feature import PCA
from pyspark.mllib.linalg import Vectors
points = parsedData.map(lambda point :
Vectors.dense(point[0:4]))
pcamod = PCA(2).fit(points)
transformed = pcamod.transform(points)
```

- K-Means
- Hierarchical
- K-Means++
- Expectation-Maximization

What is K-Means Clustering?

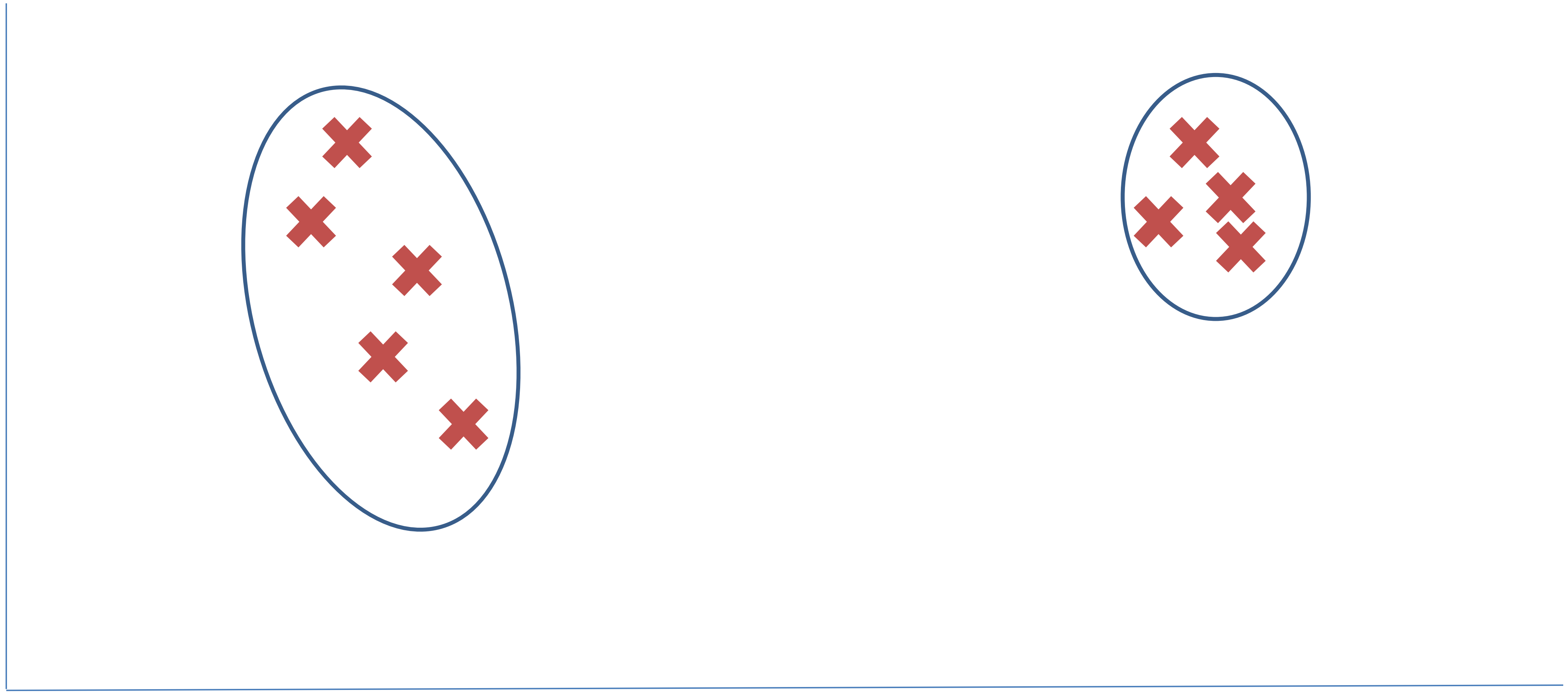


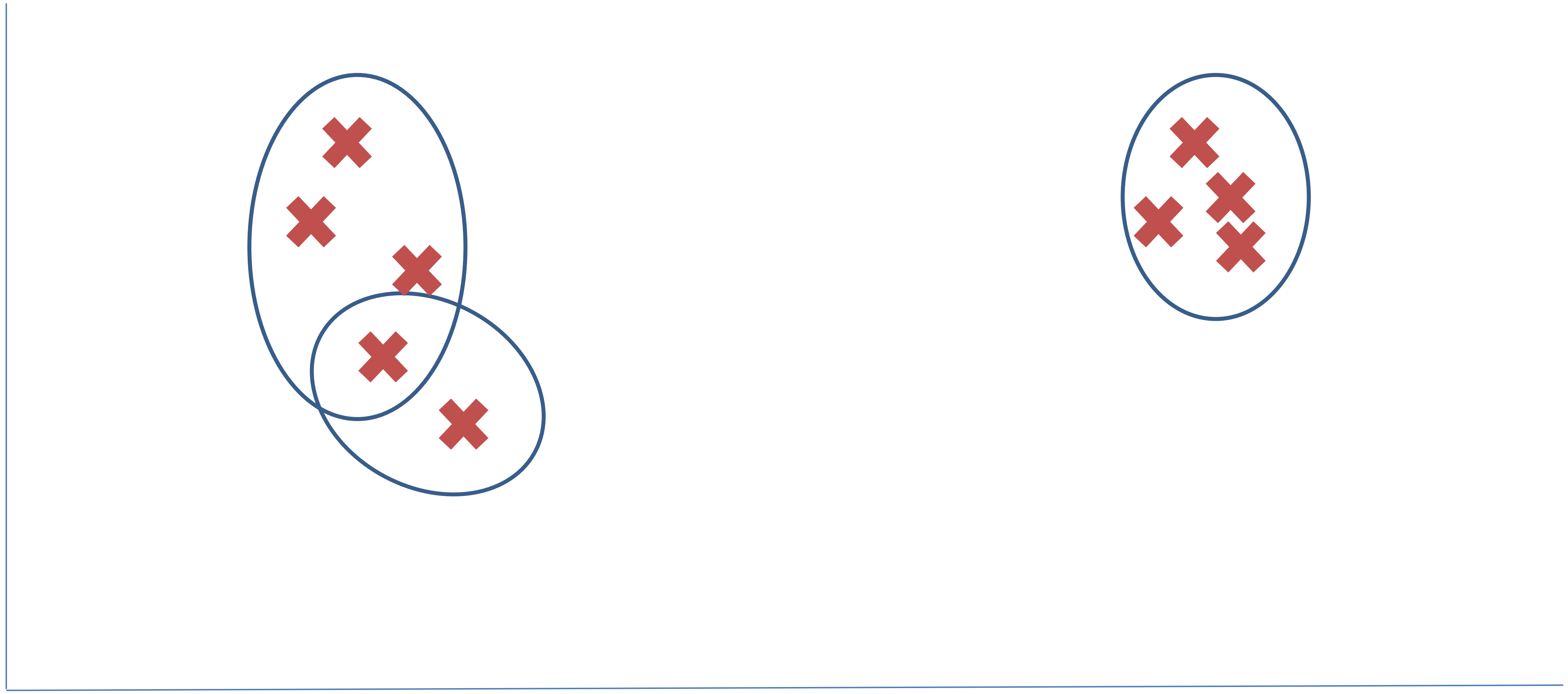
- Choose *k* points at random and form the centre of each cluster (called centroids)
- Assignment of each point to the closest cluster centre using a distance measure
- Work out the centre of the cluster and use it as the new cluster centre
- Return to the second step whilst the cluster centre changes – keep iterating until no more

- Number of centroids
- Distance Metric Type
- Initialization Type
- Number of Iterations

```
clusters = KMeans.train(transformed, 3, maxIterations=10,
runs=10, initializationMode="k-means||")

print(transformed.first())
clusters.centers
```





DEMO

Clustering data with K-Means Clustering

How do I use Spark MLLib?

- Classification
- Clustering
- Regression
- Collaborative Filtering
- Feature Extraction
- Statistics/Linear Algebra

How can I build Spark MLLib programs?

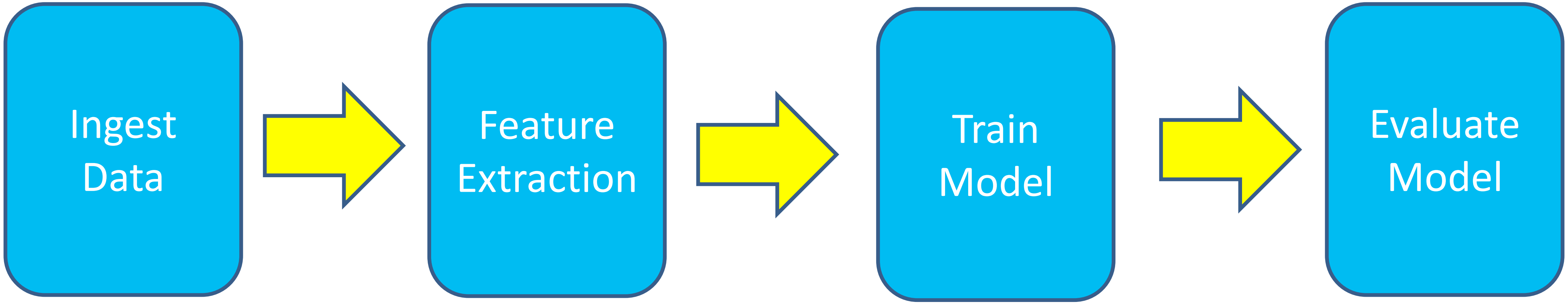
- Use a Vector type
    - Sparse
    - Dense
- LabeledPoint
- Matrix
- RowMatrix (Distributed)

- Export models as PMML
- Save models in libsvm format
- Import models from a file in libsvm format

How can I build a workflow in ML?

I think that this is a great achievement. Well done. I'm totally psyched by this. You are fantastic!

0.94



Ingest Data

Feature Extraction

Train Model

Evaluate Model

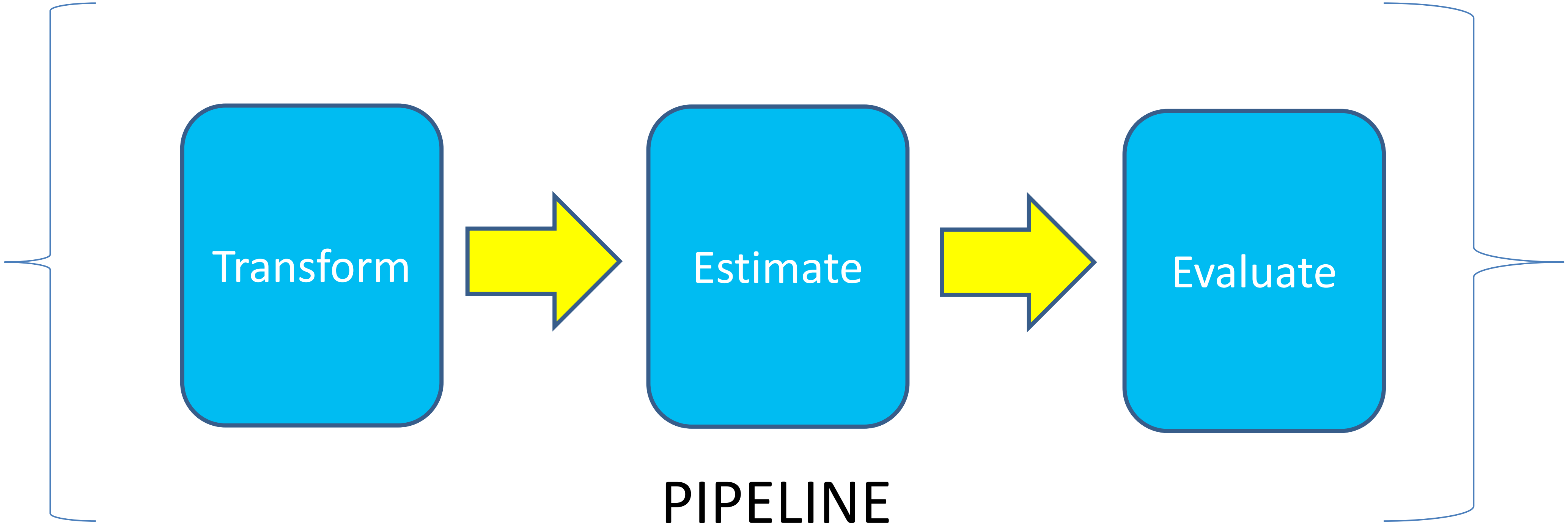
- Different ways of ingesting data and RDDs leading to messy non-reusable scripts
- Tuning hyper-parameters
- Train models for many splits of the data
- .. And for different sets of parameters

- Resilient Distributed Dataset (RDD)
- Schema based
- Domain Specific Language
- Contains named columns
- Contains types (Scala primitives)

```
case class Tweets(id: Int, label: Double, source: String, text: String)

val training = sc.textFile("/training-
tweet.csv").zipWithIndex().filter(_._2 > 0).map(line =>
line._1.split(",")).map(tw => Tweets(tw(0).toInt, tw(1).toDouble,
tw(2), tw(3))).toDF()
```

What is a pipeline?



Transform

Estimate

Evaluate

PIPELINE

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

// Fit the pipeline to training documents.
val model = pipeline.fit(training)


val modelem = model.transform(test).select("id", "label", "text",
"probability", "prediction")
```

What is a Transformer?

- DataFrame -> new DataFrame
- Extraction of values into feature vector
- Map from one column to another column
- Append an additional column
- Predict a value and append value
- Implements transform() method

```
val tokenizer = new Tokenizer().setInputCol("text").setOutputCol("words")
val hashingTF = new HashingTF().setNumFeatures(1000).setInputCol(tokenizer.getOutputCol).setOutputCol("features")
val lr = new LogisticRegression().setMaxIter(10).setRegParam(0.01)
val pipeline = new Pipeline().setStages(Array(tokenizer, hashingTF, lr))

val model = pipeline.fit(training)

val modelem = model.transform(test).select("id", "label", "text", "probability", "prediction")
```

What is an Estimator?

- Implements a method fit()
- Takes in a DataFrame as input
- Produces a Model as Output
- Model is a Transformer
- Predict a value and append value

```
val tokenizer = new Tokenizer().setInputCol("text").setOutputCol("words")
val hashingTF = new HashingTF().setNumFeatures(1000).setInputCol(tokenizer.getOutputCol).setOutputCol("features")
val lr = new LogisticRegression().setMaxIter(10).setRegParam(0.01)
val pipeline = new Pipeline().setStages(Array(tokenizer, hashingTF, lr))

val model = pipeline.fit(training)

val modelem = model.transform(test).select("id", "label", "text", "probability", "prediction")
```

- Determine how close a fit your model is to data
- Get a score to determine effectiveness of model
- Precision, recall, F-Measures
- Area Under ROC
- MSE/RMSE

DEMO

Building a Spark ML Pipeline

- What is Binary Classification? What is Multiclass Classification?
- What is regression?
- What is collaborative filtering?
- What is Unsupervised Learning?
- What is K-Means Clustering?
- How do I use Spark MLLib?
- How to I build Spark ML programs?
- How can I build a workflow in ML?
- What is a pipeline?
- What is a Transformer?
- What is an Estimator?

Microsoft