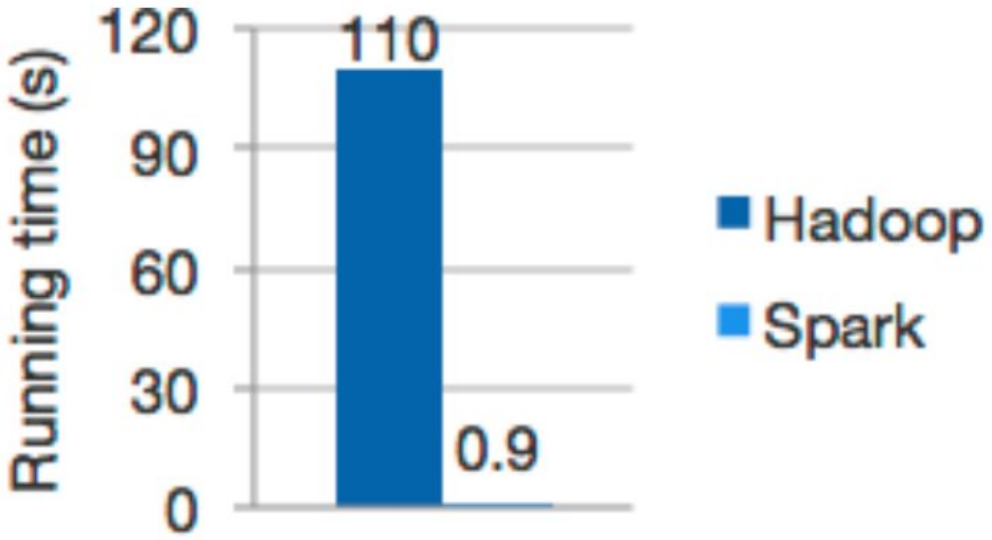
SparkML basics

Dmitry Bugaychenko

@mail.ru group



Logistic regression in Hadoop and Spark

History of Spark APIs

RDD (2011) → DataFrame (2013) → DataSet (2015)

RDD (2011)

Distribute collection of JVM objects

Functional Operators (map, filter, etc.)

DataFrame (2013)

Distribute collection of Row objects

Expression-based operations and UDFs

Logical plans and optimizer

Fast/efficient internal representations

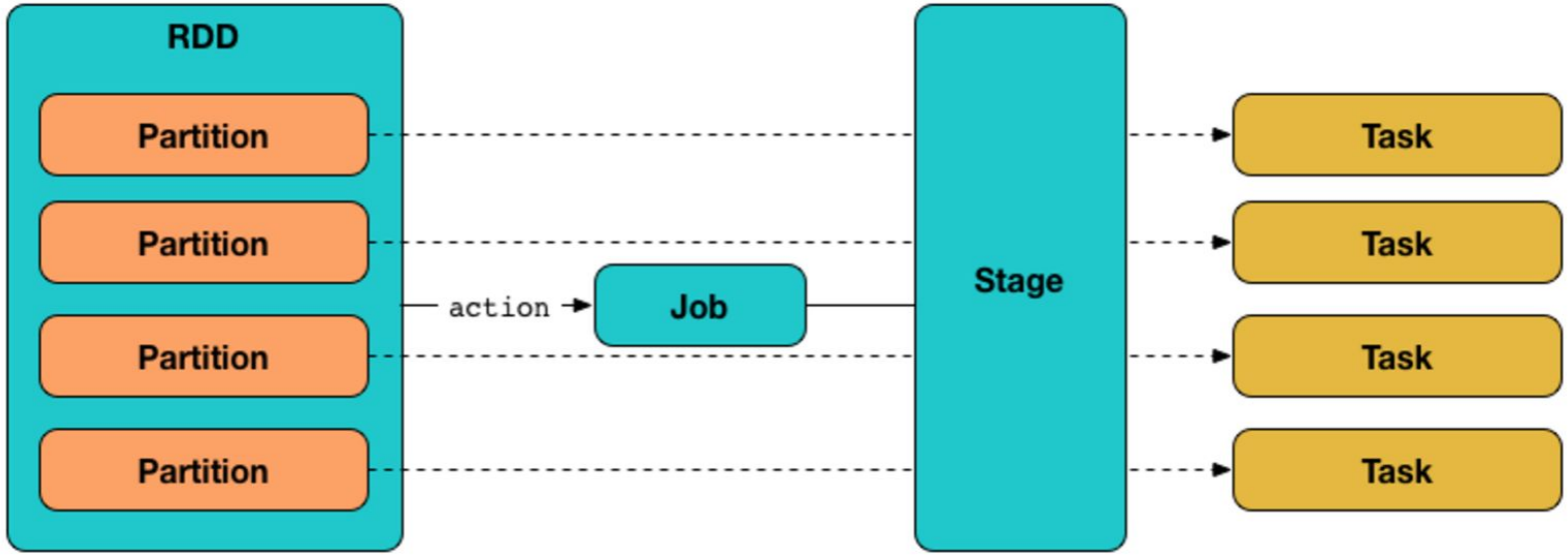
DataSet (2015)

Internally rows, externally JVM objects

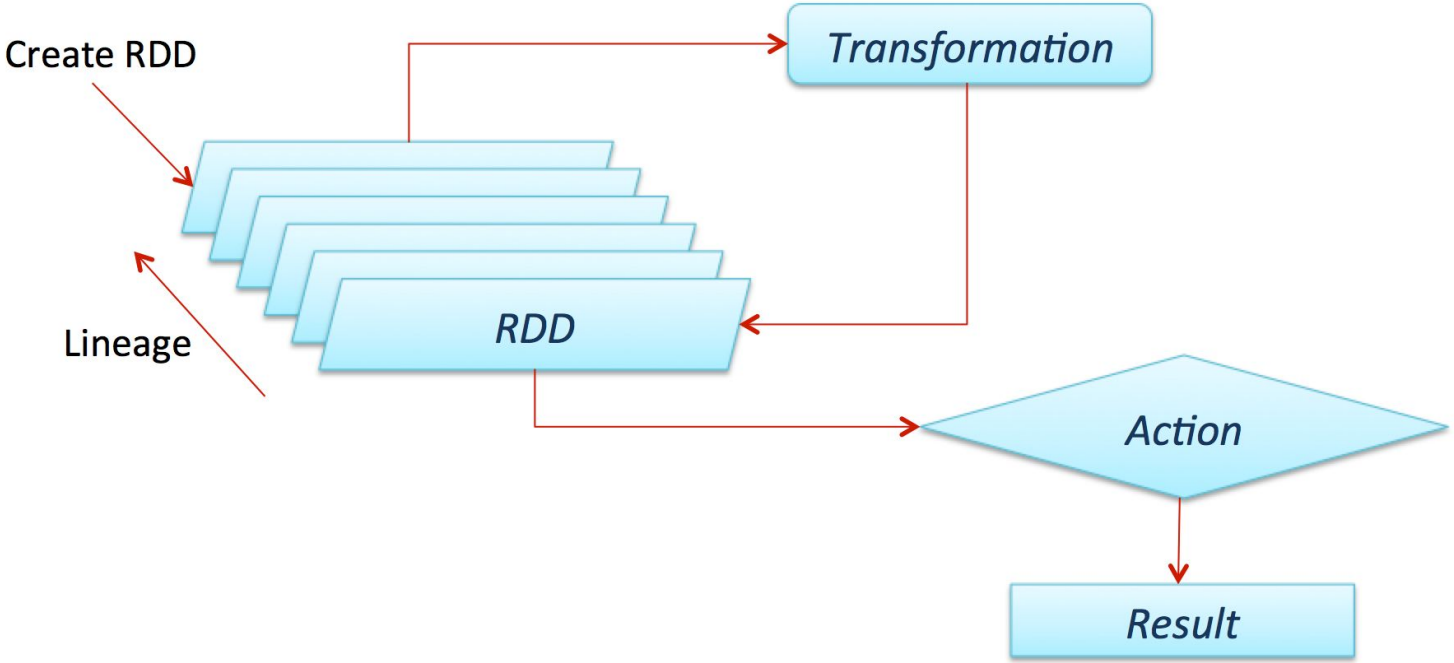
Almost the "Best of both worlds": type safe + fast

But slower than DF
Not as good for interactive analysis, especially Python

RDD Basics

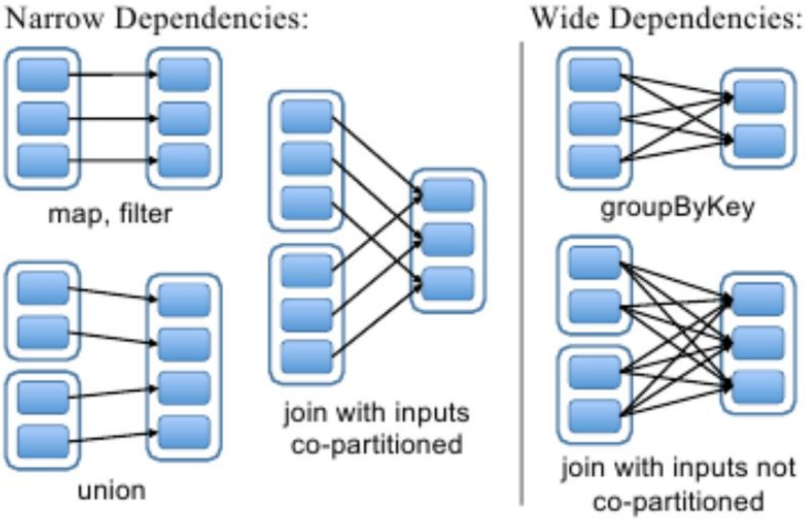

RDD
Partition
Partition
Partition
Partition
action
Job
Stage
Task
Task
Task
Task

RDD Basics



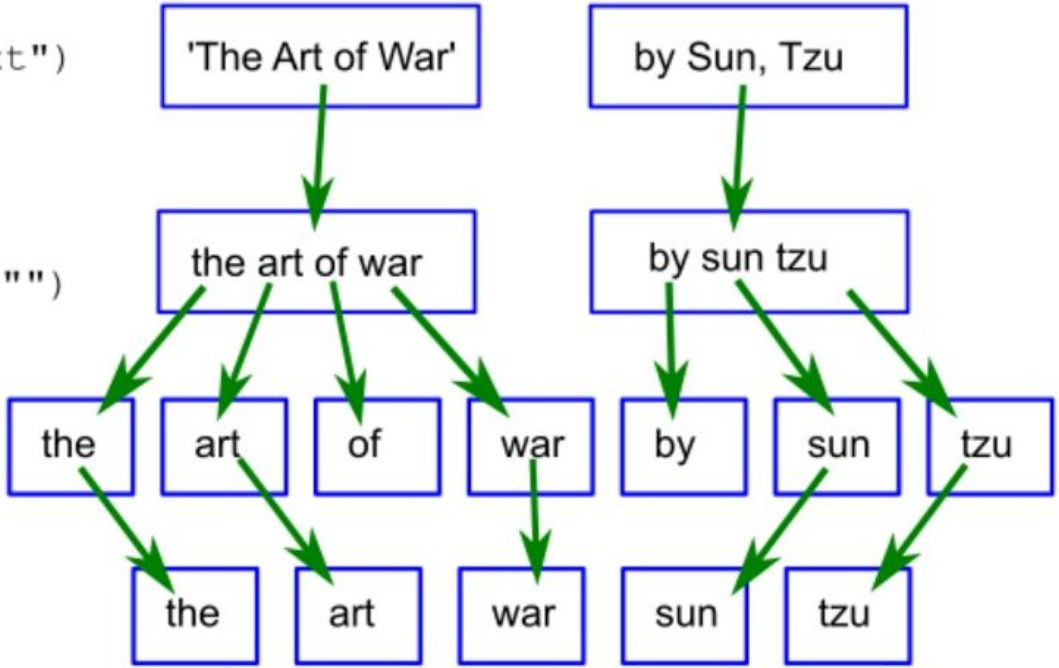
RDD Basics

Transformations	Actions
map(func)	take(N)
flatMap(func)	count()
filter(func)	collect()
groupByKey()	reduce(func)
reduceByKey(func)	takeOrdered(N)
mapValues(func)	top(N)
...	...

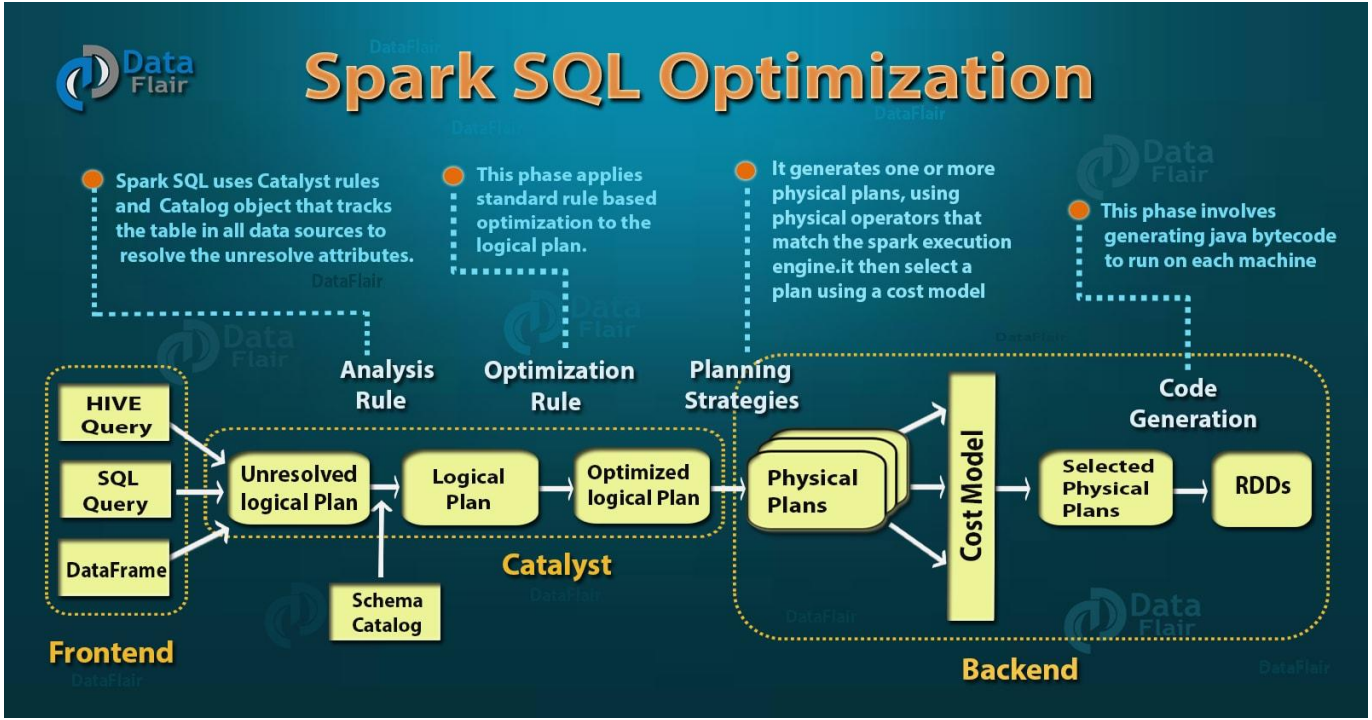


RDD Basics

```
sc.textFile("artofwar.txt")

.map(
  _.toLowerCase
  .replaceAll("[^\\w ], "")

 .flatMap(_.split(" "))

   .filter(_.size >2)
```



DataFrames


Data Flair
Spark SQL Optimization
Spark SQL uses Catalyst rules and Catalog object that tracks the table in all data sources to resolve the unresolve attributes.
This phase applies standard rule based optimization to the logical plan.
It generates one or more physical plans, using physical operators that match the spark execution engine.it then select a plan using a cost model
This phase involves generating java bytecode to run on each machine
Analysis Rule
Optimization Rule
Planning Strategies
Code Generation
HIVE Query
SQL Query
DataFrame
Unresolved logical Plan
Logical Plan
Optimized logical Plan
Physical Plans
Cost Model
Selected Physical Plans
RDDs
Schema Catalog
Catalyst
Frontend
Backend

Datasets

- DataFrame (with relational operations) and Dataset (with lambda functions) use Catalyst and row-oriented data representation on off-heap

```
case class Pt(x: Int, y: Int)
d = Array(Pt(1, 4), Pt(2, 5))
```

DataFrame (v1.3-)

```
df = d.toDF(…)
df.filter("x>1")
  .count()
```

Dataset (v1.6-)

```
ds = d.toDS()
ds.filter(p => p.x>1)
  .count()
```

RDD (v0.5-)

```
rdd = sc.parallelize(d)
rdd.filter(p => p.x>1)
  .count()
```

Frontend API

Catalyst

Generated Java bytecode

Java bytecode in Spark program and runtime

Backend computation

Row-oriented

Data

IBM

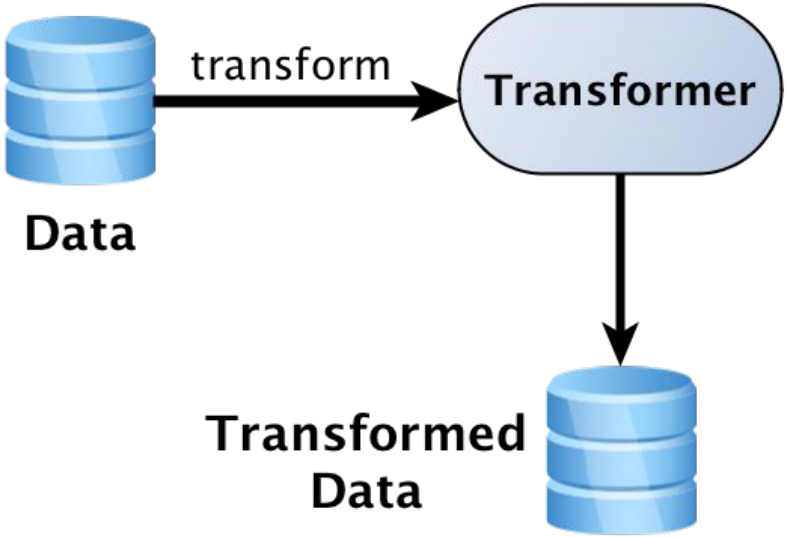
SQL vs. DataFrame vs. Dataset

	SQL	DataFrames	Datasets
Syntax Errors	Runtime	Compile Time	Compile Time
Analysis Errors	Runtime	Runtime	Compile Time

Spark ML Pipelines

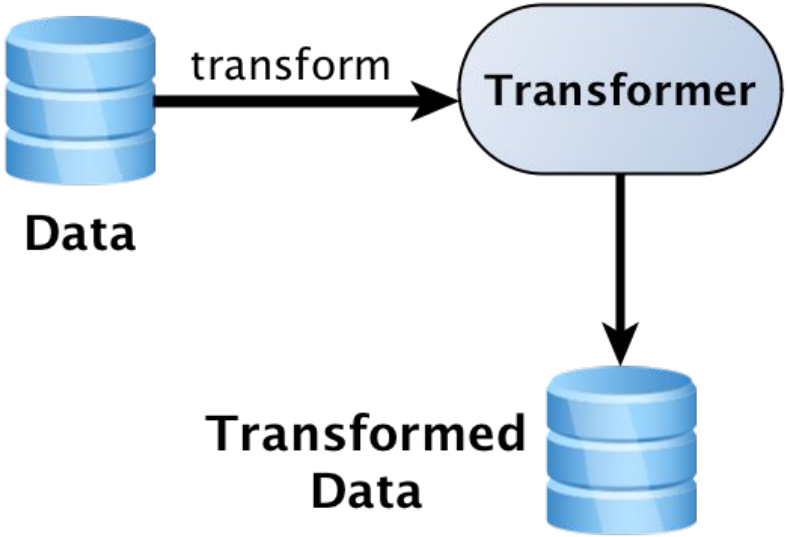
Spark ML Pipelines

Transformer

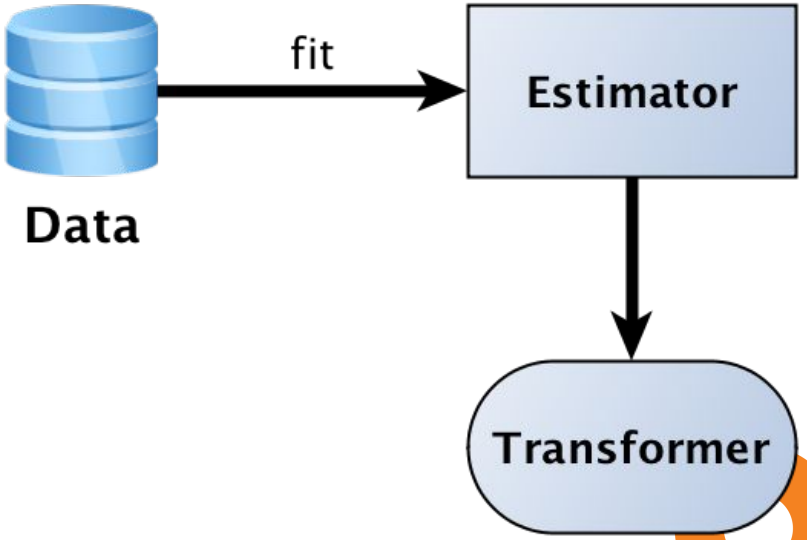


Spark ML Pipelines

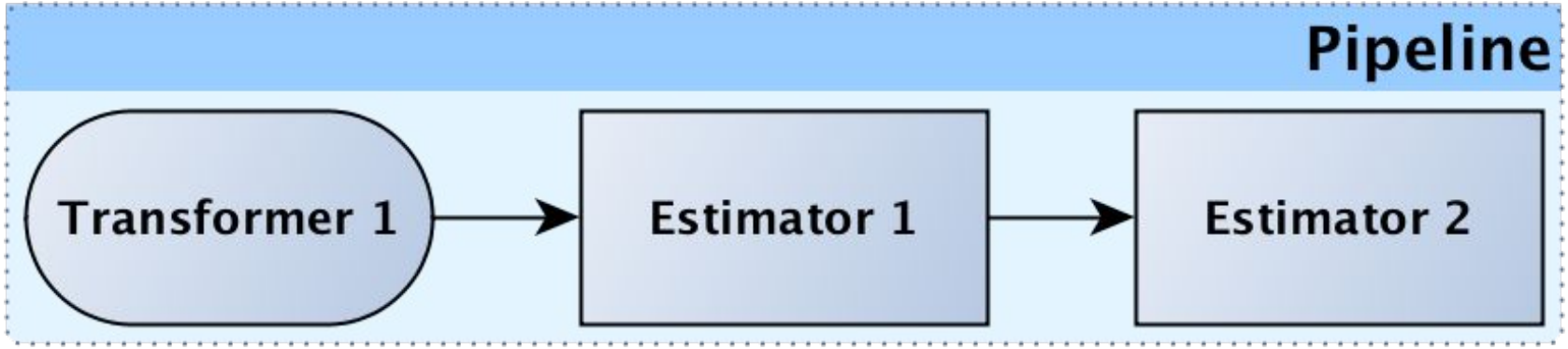
Transformer



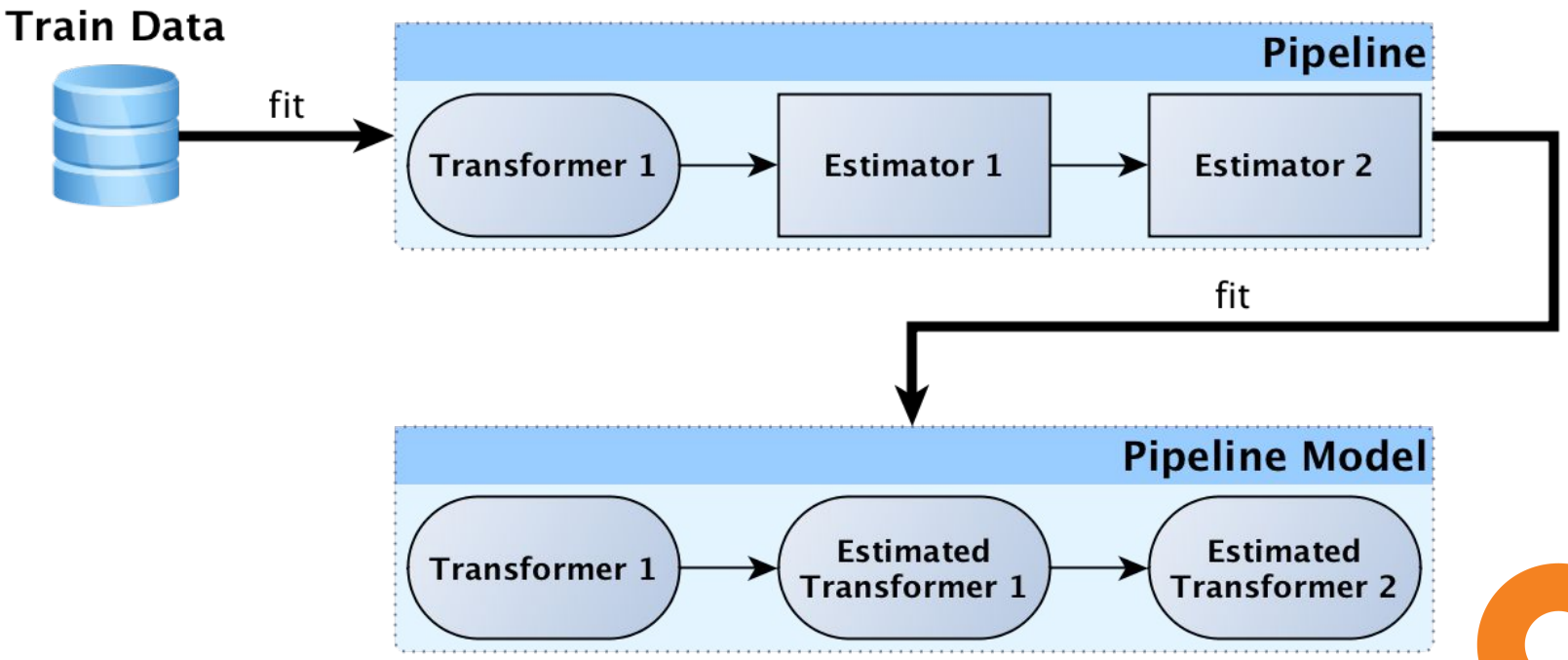
Estimator



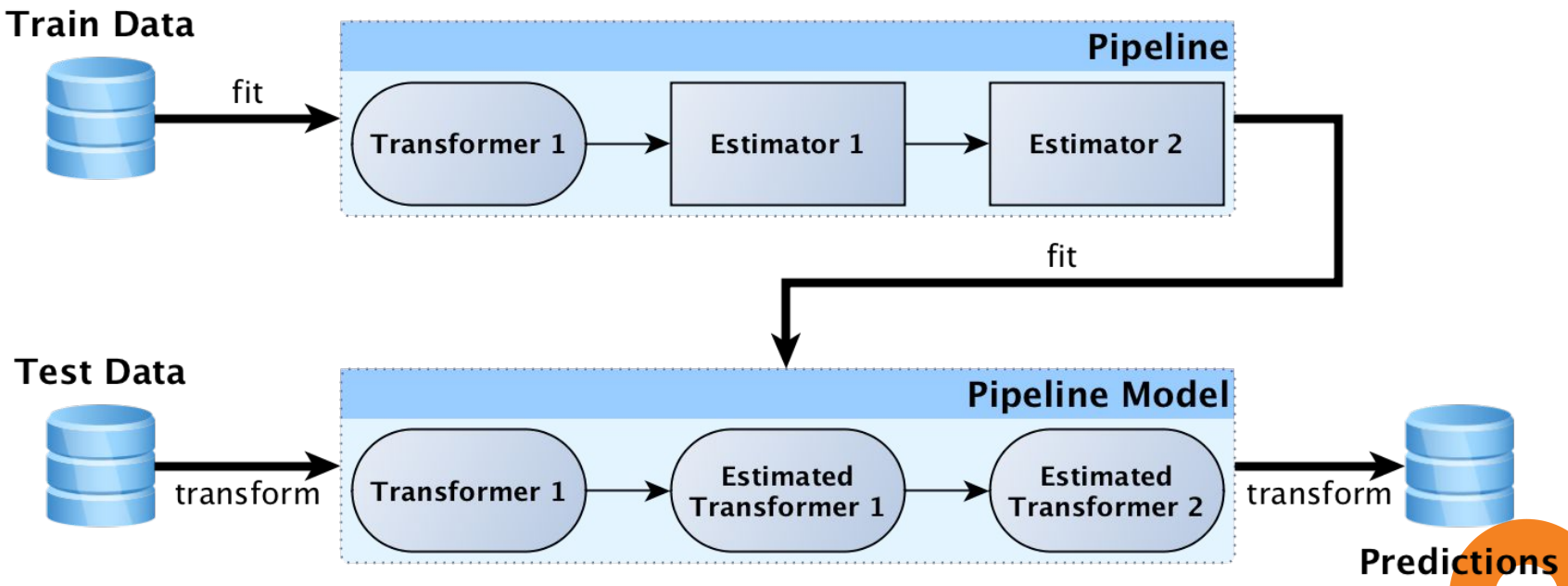
Spark ML Pipelines



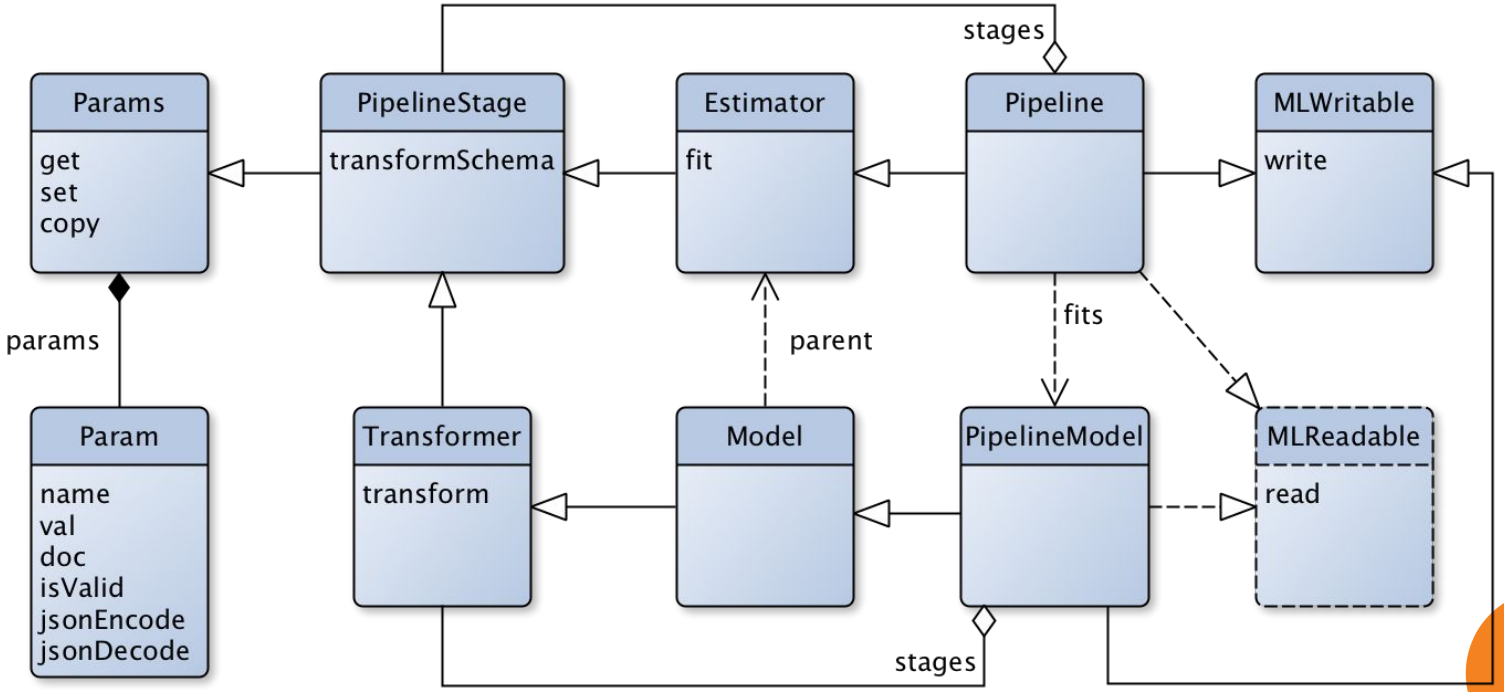
Spark ML Pipelines



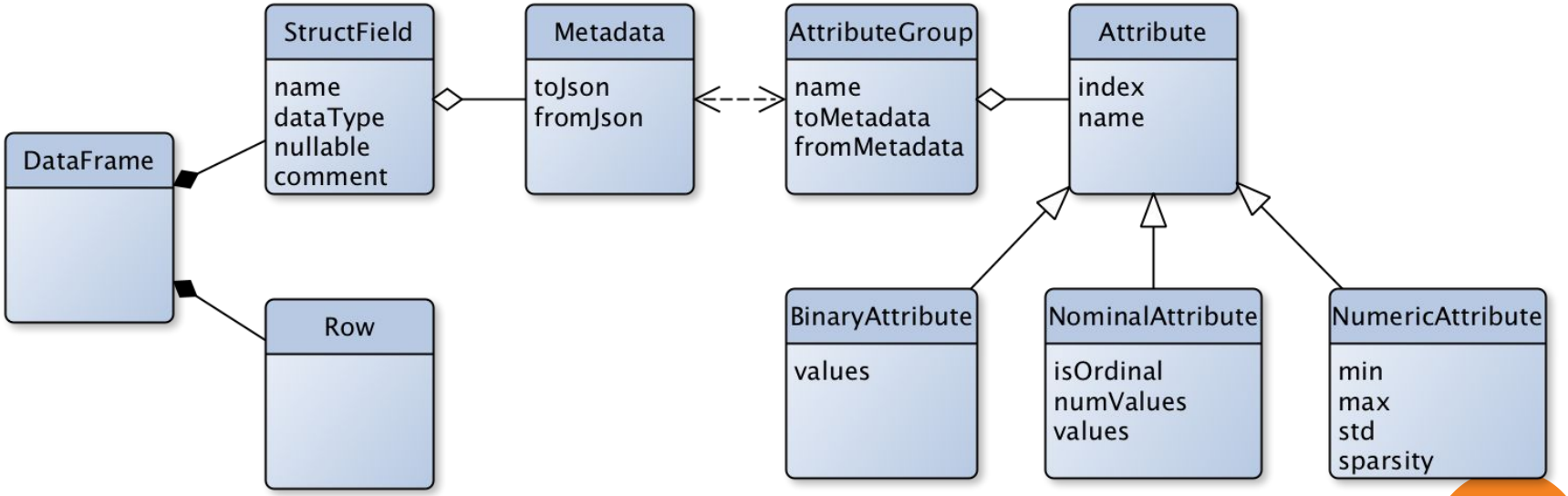
Spark ML Pipelines



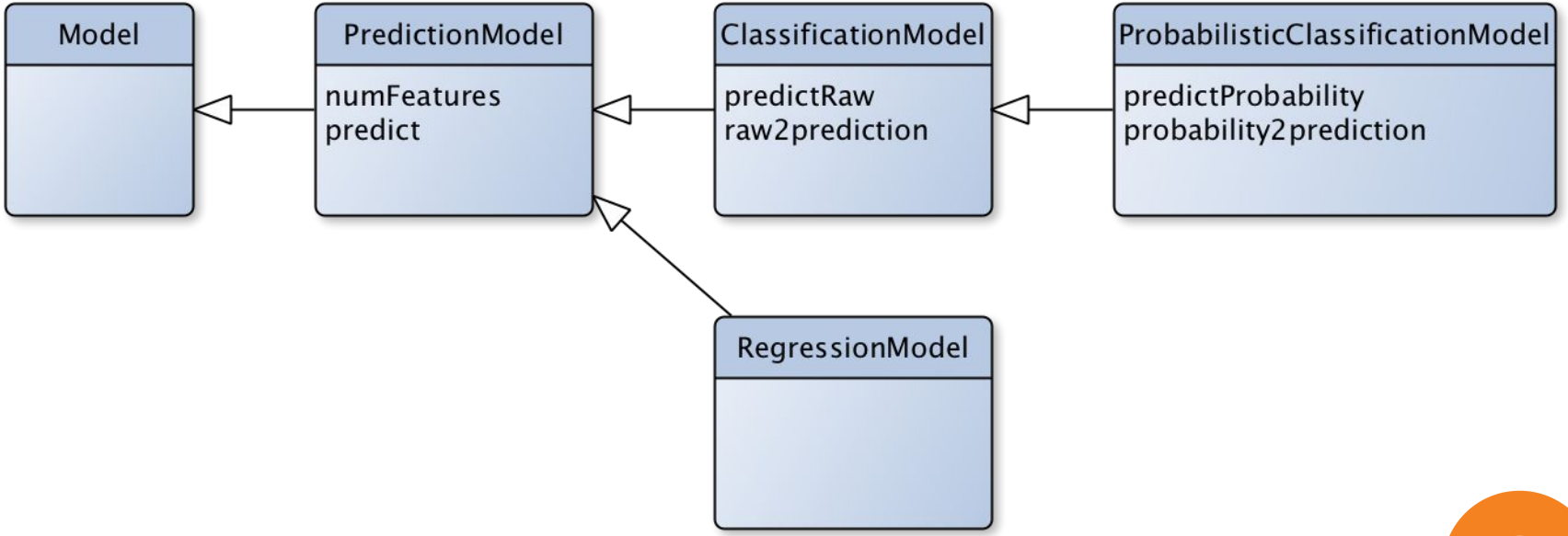
Spark ML Core



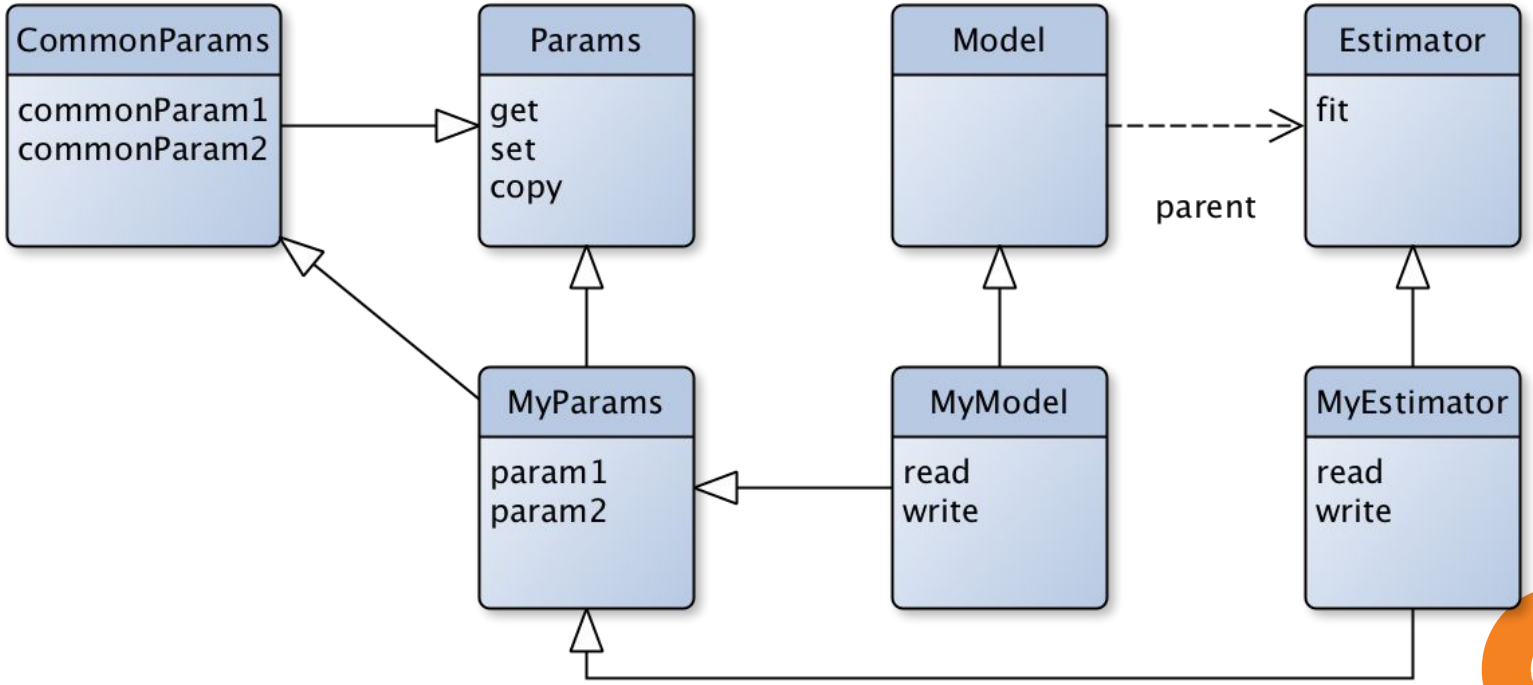
Field Metadata and Attributes



Prediction Model



“My Spark ML Model”



Spark ML Features

- ETL
  - SQLTransformer
  - *SqlFilter, ColumnsExtractor*
- Numerization
  - OneHotEncoder
  - StringIndexer
  - *MultinomialExtractor*
- Vectorization
  - VectorAssembler
  - FeatureHasher
  - *AutoAssembler*
- Feature Normalization
  - MaxAbsScaler
  - MinMaxScaler
  - Normalizer
  - QuantileDiscretizer
  - StandardScaler
- Missing values
  - Imputer
  - *NullToDefaultReplacer*
  - *NaNToMeanReplacer*

Spark ML Features

- Feature Engineering
  - DCT
  - ElementwiseProduct
  - Interaction
  - VectorIndexer
  - PolynomialExpansion
- Feature Selection
  - ChiSqSelector
  - *FoldedFeaturesSelector*
- Dimension reduction
  - PCA
  - MinHashLSHModel
  - BucketedRandomProjectionLSH
  - *RandomProjectionsHasher*

Spark ML Features

- Texts extraction
  - Tokenizer
  - RegexTokenizer
  - Ngram
  - StopWordsRemover
- NLP in Pravada-ML
  - *LanguageDetectorTransformer*
  - *LanguageAwareAnalyzer*
  - *NGramExtractor*
  - *URLElimminator*
- Texts vecotization
  - CountVectorizer
  - HashingTF
  - IDF
- Text embedding
  - Word2Vec
- Clustering
  - LDA
  - KMeans/BisectingKMeans
  - GaussianMixture

Spark ML Features

Regression

- AFTSurvivalRegression.scala
- DecisionTreeRegressor.scala
- GBTRegressor.scala
- GeneralizedLinearRegression.scala
- IsotonicRegression.scala
- LinearRegression.scala
- RandomForestRegressor.scala

Classification

- DecisionTreeClassifier.scala
- GBTClassifier.scala
- LinearSVC.scala
- LogisticRegression.scala
- MultilayerPerceptronClassifier.scala
- NaiveBayes.scala
- OneVsRest.scala
- ProbabilisticClassifier.scala
- RandomForestClassifier.scala

Spark ML Features

- Recommendations
  - ALS
  - FPGrowth
- Evaluation
  - BinaryClassificationEvaluator
  - ClusteringEvaluator
  - MulticlassClassificationEvaluator
  - RegressionEvaluator
- Tuning
  - ParamGridBuilder
  - CrossValidator
- More from Pravda-ML
  - *CombinedModel*
  - *PartitionedRankingEvaluator*
  - *CRRSampler*
  - *XGBoost*
  - *StochasticHyperopt*

Thank you for your attention!

Dmitry.Bugaychenko
@corp.mail.ru