

APACHE SPARK INTRODUCTION

By Véronique Mulholland

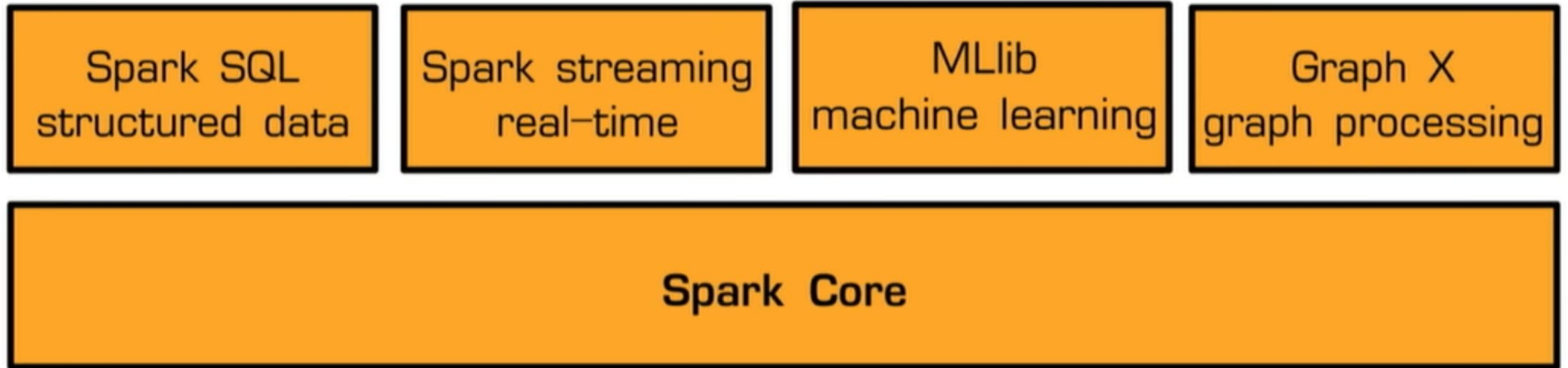
Data Manager @ Quebec Iron Ore

2018.11.20

APACHE SPARK

Fast and general engine for large-scale data processing and cloud computing

APACHE SPARK ECOSYSTEM



DISTRIBUTED COMPUTE

Fast and general engine for large-scale
data processing

RESILIENT DISTRIBUTED DATASET (RDD)

Collection of elements partitioned across
the nodes of the cluster that can be
operated on in parallel

RDD FEATURES

- **Distributed collection**
- **Immutable**
- **Fault tolerance**
- **Lazy evaluations**
- **Functional transformations**
- **Data processing formats**
- **RDD API is available in Java, Scala, Python and R**
- **Spark.mllib class is built on top of RDD**

RDD LIMITATIONS

- **No inbuilt optimization engine**
- **Handling structured data**

DATAFRAME/DATASET

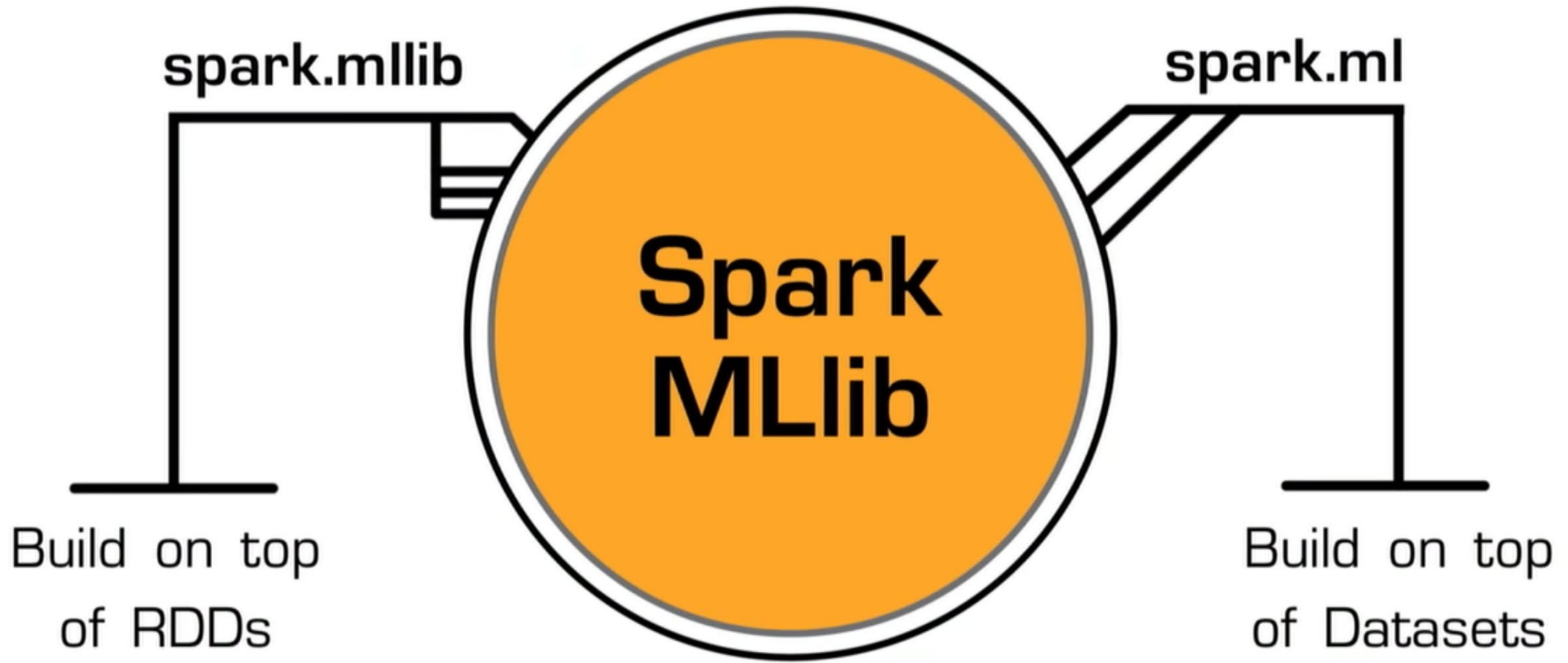
**Immutable distributed collection of data
that data is organized into a **table with
named columns****

DATAFRAME/DATASET FEATURES

- **Distributed collection of Row Object**
- **Data Processing**
- **Optimization using catalyst optimizer**
- **Hive Compatibility**
- **Tungsten**
- **Dataframe API is available in Java, Scala, Python and R**
- **More modern spark.ml class**

DATAFRAME/DATASET LIMITATIONS

- **Compile-time type safety**
- **Cannot operate on domain Object (lost domain object)**



SPARK.MLLIB FEATURES

- **Utilities: linear algebra, statistics, etc.**
- **Feature extraction, features transformations, etc.**
- **Regression**
- **Classification**
- **Clustering**

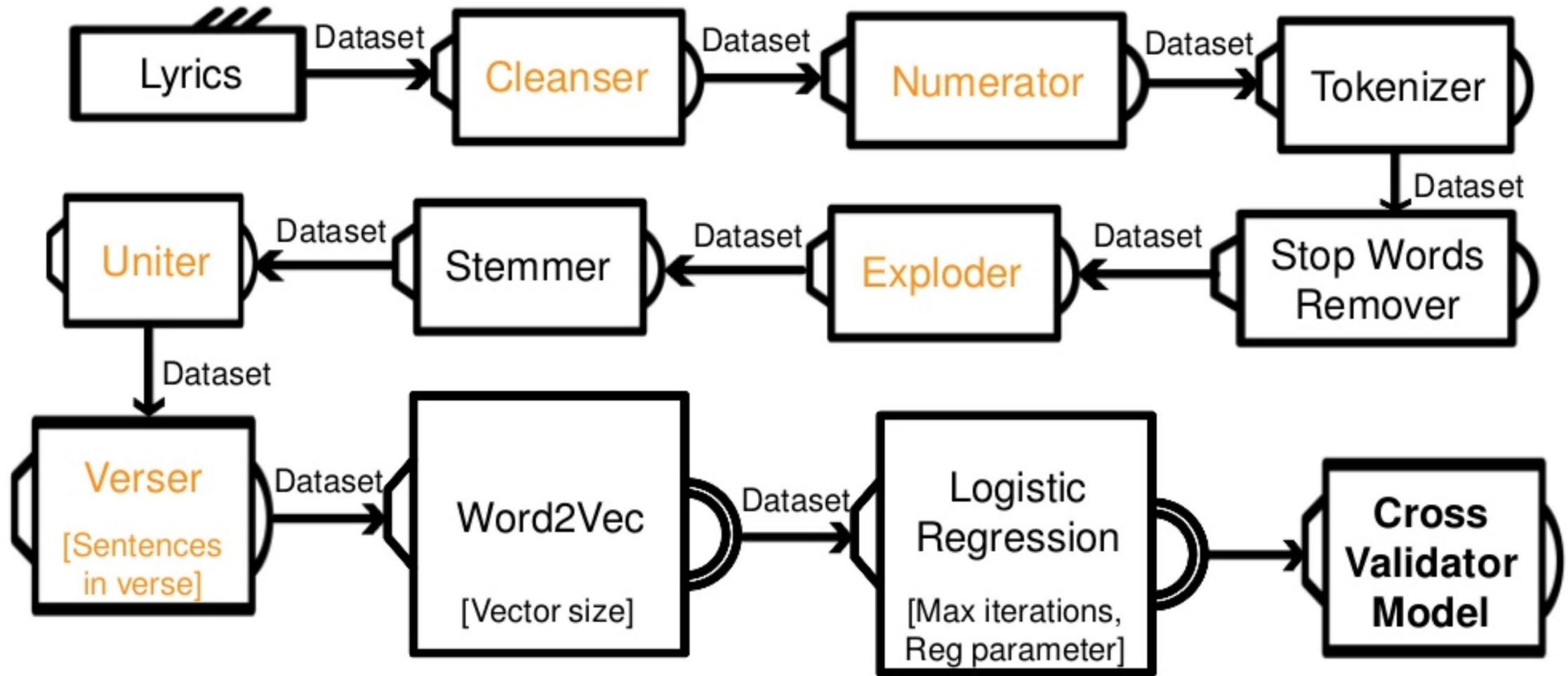
SPARK.ML FEATURES

- Utilities: linear algebra, statistics, etc.
- Feature extraction, features transformations, etc.
- Regression
- Classification
- Clustering
- **Pipelines with persistence**
- **Model Selection and tuning**
 - Train/validate/split
 - K-fold cross validation



Inspired by Python's SKLearn

EXAMPLE SPARK ML PIPELINE



SPARK ML PIPELINE EXAMPLE

```
// Create model.
Word2Vec word2Vec = new Word2Vec().setInputCol(VERSE.getName()).setOutputCol("features").setMinCount(0);

LogisticRegression logisticRegression = new LogisticRegression();

Pipeline pipeline = new Pipeline().setStages([
    new PipelineStage[]{
        cleanser,
        numerator,
        tokenizer,
        stopWordsRemover,
        exploder,
        stemmer,
        uniter,
        verser,
        word2Vec,
        logisticRegression});

// Use a ParamGridBuilder to construct a grid of parameters to search over.
ParamMap[] paramGrid = new ParamGridBuilder()
    .addGrid(verser.sentencesInVerse(), new int[]{4, 8, 16})
    .addGrid(word2Vec.vectorSize(), new int[] {100, 200, 300})
    .addGrid(logisticRegression.regParam(), new double[] {0.01D})
    .addGrid(logisticRegression.maxIter(), new int[] {100, 200})
    .build();
```

PYTHON SKLEARN PIPELINE EXAMPLE

```
lm = Lemmatizer()
tfidf = TfidfVectorizer(max_features=max_features)
lr = LogisticRegression()
nb = NBFeaturer(1)
p = Pipeline([
    ('lm', lm),
    ('tfidf', tfidf),
    ('nb', nb),
    ('lr', lr)
])

cross_val_score(estimator=p, X=x, y=y, scoring=scoring, cv=cv, n_jobs=n_jobs)
```

SUMMARY

Why Move to Apache Spark?

- **Swift Processing**
- **Dynamic in Nature**
- **In-Memory Computation**
- **Re-Usability**
- **Fault Tolerance**

Why Admire from Afar?

- **Expensive**
- **Latency**
- **Manual Optimization**
- **No File Management**
- **Problem With Small Files**

INTRODUCTORY SPARK USES CASES

- Link 1: [HealthCare Use Case With Apache Spark](#)
- Link 2: [Introduction to Spark RDD and Basic Operations in RDD](#)
- Link 3: [Analyzing New York Crime Data Using SparkSQL](#)
- Link 4: [Spark Use Case – Travel Data Analysis](#)
- Link 5: [Spark Use Case – Uber Data Analysis](#)
- Link 6: [Spark Use Case – Analyzing MovieLens Dataset](#)
- Link 7: [Spark Use Case – Social Media Analysis](#)