



University of  
South Australia

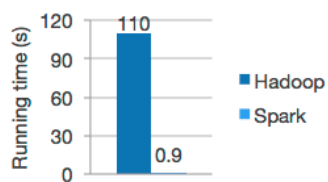
## Machine learning in Spark

1

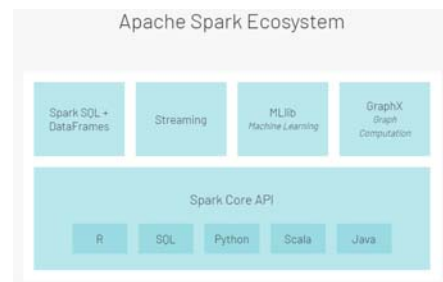


MLlib fits into Spark's APIs and interoperates with NumPy in Python (as of Spark 0.9) and R libraries (as of Spark 1.5). You can use any Hadoop data source (e.g. HDFS, HBase, or local files), making it easy to plug into Hadoop workflows.

As of Spark 2.0 MLlib has switched to dataFrames as the primary API.



Logistic regression in Hadoop and Spark



2

## MLlib is based on scikit-learn

Scikit-learn is a library in Python for machine learning, which is built on NumPy, SciPy and matplotlib.

Scikit-learn is based around the idea of 'pipelines' and MLlib is too.

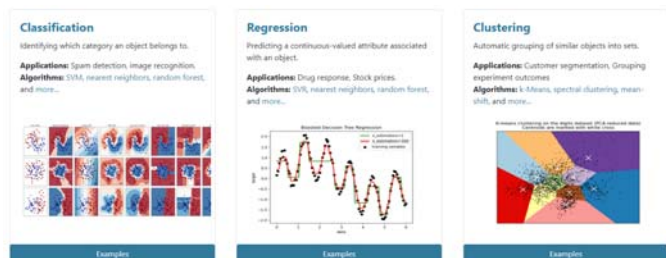
**scikit-learn**

*Machine Learning in Python*

Getting Started

Release Highlights for 0.23

GitHub



## What are pipelines

The idea behind pipelines is to be able to combine multiple algorithms into one machine learning 'pipeline'. There are a number of components that make it work.

- dataFrames

## What are pipelines

The idea behind pipelines is to be able to combine multiple algorithms into one machine learning 'pipeline'. There are a number of components that make it work.

- dataFrames
- Transformers

```
def first_append(df)
    return df + column('hello')

def second_append(df)
    return df + column('there')

new_dF = old_dF.transform(first_append).transform(second_append)
```

## What are pipelines

The idea behind pipelines is to be able to combine multiple algorithms into one machine learning 'pipeline'. There are a number of components that make it work.

- dataFrames
- Transformers
- Estimators

## What are pipelines

The idea behind pipelines is to be able to combine multiple algorithms into one machine learning 'pipeline'. There are a number of components that make it work.

- dataFrames
- Transformers
- Estimators
- Pipeline

## What are pipelines

The idea behind pipelines is to be able to combine multiple algorithms into one machine learning 'pipeline'. There are a number of components that make it work.

- dataFrames
- Transformers
- Estimators
- Pipeline

### Linear regression in MLlib

Build the model

```
# Import LinearRegression class
from pyspark.ml.regression import LinearRegression

# Define LinearRegression algorithm
lr = LinearRegression()

# Fit 2 models, using different regularization parameters
modelA = lr.fit(data, {lr.regParam:0.0})
modelB = lr.fit(data, {lr.regParam:100.0})
```

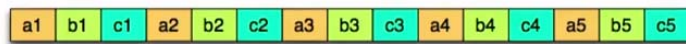
<https://databricks.com/spark/getting-started-with-apache-spark/machine-learning>

# Output in Spark - Parquet

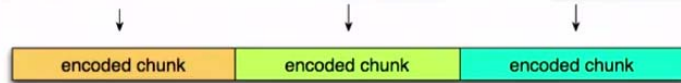
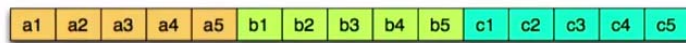
Logical table representation

a	b	c
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b4	c4
a5	b5	c5

Row layout



Column layout



[https://www.youtube.com/watch?v=dPb2ZXnt2\\_U](https://www.youtube.com/watch?v=dPb2ZXnt2_U)

## WARNING

This material has been reproduced and communicated to you by or on behalf of the **University of South Australia** in accordance with section 113P of the *Copyright Act 1968 (Act)*.

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

**Do not remove this notice**