University of South Australia

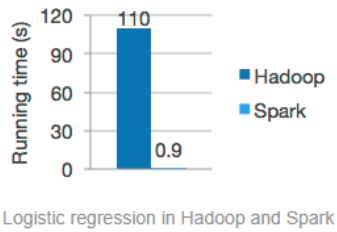
Machine learning in Spark

1

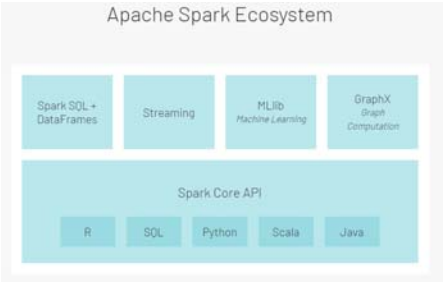
MLlib

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



University of South Australia

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.

3

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

4

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)
```


5

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

6

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

7

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

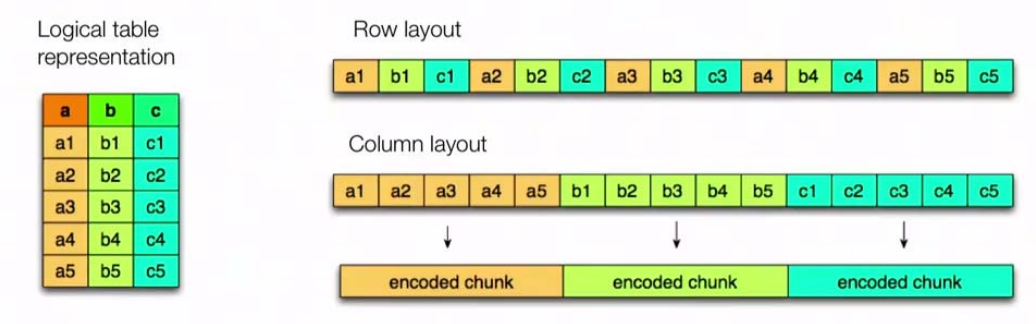
Build the model

```
# 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

8

Output in Spark - Parquet



https://www.youtube.com/watch?v=dPb2ZXnt2_U